

Utveckling av ny webbplats för Svenska Handelshögskolans Studentkår med Drupal 7

Jonas Engström

Examensarbete
Informations- och medieteknik
2014

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informations- och medieteknik
Identifikationsnummer:	4697
Författare:	Jonas Rainer Engström
Arbetets namn:	Utveckling av ny webbplats för Svenska Handelshögskolans Studentkår med Drupal 7
Handledare (Arcada):	Johnny Biström
Uppdragsgivare:	Oy Nord Software Ltd
<p>Sammandrag:</p> <p>Examensarbetet handlar om utvecklingen av en ny webbplats för Svenska Handelshögskolans Studerandekår. Den nya webbplatsen ska ge möjlighet för sidadministratörer att skapa och moderera bloggar, evenemang, materialbanker, gallerier, forum och statiska sidor. Sidan är även ämnad att vara tvåspråkig och även delas upp på två campus. Webbplatsen byggs med Drupal 7 som är ett PHP-baserat innehållshanteringssystem.</p> <p>I arbetet byggs en grunduppfattning om Drupal 7 som innehållshanteringssystem upp med en genomgång av de element som används vid innehållsstrukturering och Drupal-utveckling. Därefter redogörs för hur ett Drupal-tema är uppbyggt genom forskning i ett temas struktur och modifikationsprocesser. Den praktiska delen av arbetet klargör utvecklingsprocessen för webbplatsens innehållsstruktur och dess presentationslösningar. Redogörelsen av utvecklingsprocessen byggs upp på kunskap från teoridelen.</p> <p>Arbetet ämnar inte förklara utvecklingen från installation till färdig produkt utan snarare redogöra för den strukturella utvecklingen av en webbplats baserad på Drupal 7.</p>	
Nyckelord:	Drupal, Innehållshanteringssystem, Nod, Modul, Förprocessering, Teman, Mallar, Flerspråkighet
Sidantal:	66
Språk:	Svenska
Datum för godkännande:	23.4.2014

DEGREE THESIS	
Arcada	
Degree Programme:	Information- and media technology
Identification number:	4697
Author:	Jonas Engström
Title:	Development of a new website for the Student Union of Hanken School of Economics with Drupal 7
Supervisor (Arcada):	Johnny Biström
Commissioned by:	Oy Nord Software Ltd.
<p>Abstract:</p> <p>This degree thesis tells of the development of a new website for the Student Union of Hanken School of Economics. The website will support the site administrators in creating and moderating blogs, events, material collections, galleries, forums and static pages. The website is also created for international purposes with two main languages and also a splitting of the site material for two campuses. The website is to be built with Drupal 7 which is a content management system built on PHP.</p> <p>In this thesis a basic understanding of Drupal 7 as a content management system is achieved with a review of the elements that are used in the creation of a content structure and in Drupal development. Following that the structure of a Drupal theme is explained through researching it and the customization processes. The practical part of the thesis is to clarify the development process of the content structure and the solutions for its presentation. The explanation of the development process builds on the knowledge attained in the theoretical part of the thesis.</p> <p>This degree thesis does not attempt to explain the development process from installation to publication but rather explain the structural development of a Drupal 7 based website.</p>	
Keywords:	Drupal, Content management system, Node, Module, Pre-processing, Themes, Templates, Multilingual
Number of pages:	66
Language:	Swedish
Date of acceptance:	23.4.2014

INNEHÅLL

1	Inledning.....	8
1.1	Utgångspunkt	8
1.2	Målsättning	8
1.3	Metoder	8
1.4	Begränsningar	9
2	Colangine-temat.....	9
3	Drupal 7	11
3.1	Drupals struktur	11
3.1.1	<i>Noder</i>	12
3.1.2	<i>Block</i>	12
3.1.3	<i>Menyer</i>	12
3.1.4	<i>Taxonomi</i>	13
3.2	Moduler.....	13
4	Temautveckling i Drupal 7.....	14
4.1	Vad är ett tema	14
4.2	Temats uppbyggnad.....	14
4.2.1	<i>Filen .info</i>	16
4.2.2	<i>Mallar</i>	18
4.2.3	<i>Ersättning av Drupals temafunktioner</i>	19
4.2.4	<i>Förprocessering</i>	20
4.2.5	<i>CSS- och Javascript-filer</i>	21
4.2.6	<i>Temarelaterat bildmaterial</i>	22
5	Utveckling av SHS.fi	22
5.1	Grundimplementation av Colangine-temat.....	23
5.1.1	<i>Sidhuvud och sidfot</i>	24
5.1.2	<i>Implementering av rullgardinsmeny</i>	25
5.2	Innehållsstruktur	27
5.2.1	<i>Basic page</i>	28
5.2.2	<i>Klubbar, organ och utskott</i>	30
5.2.3	<i>Blogg</i>	37
5.2.4	<i>Evenemang</i>	43
5.2.5	<i>Galleri</i>	46
5.2.6	<i>Forum</i>	49
5.2.7	<i>Tentbibliotek</i>	50

5.2.8	<i>Banners</i>	52
5.3	Startsidan	52
5.4	Aktuellt-sidans struktur	56
5.5	Lokalisation.....	58
5.5.1	<i>Språk</i>	59
5.5.2	<i>Campus</i>	60
6	Diskussion	62
6.1	Drupals temamöjligheter	63
6.2	Innehållstrukturen	64
6.3	Slutsats	64
Källor	65

Figurer

Figur 1. Colangine - Creative Flat HTML5 Template produktbeskrivning (Themeforest.net 2014)	10
Figur 2. Beskrivning av strukturen i ett Drupal 7 tema (Drupal.org 2014a)	15
Figur 3. Temakatalog för Colangine temat.....	16
Figur 4. colangine.info-filen för Colangine- temat.....	17
Figur 5. HTML-märkspråk med PHP-sekvenser.....	18
Figur 6. Drupal-temat Colangines html.tpl.php-fil byggd med HTML5-temat Colangine som grund	19
Figur 7. Ersättning av temafunktionen theme_breadcrumb	20
Figur 8. Förprocesseringsfunktion för kommentarer.....	21
Figur 9. Det modifierade <body>-elementet	24
Figur 10. Utskrift av meny med funktionen theme	25
Figur 11. Ersättning av funktionen theme_menu_tree för att ge -taggen den saknade klassen sf-menu	26
Figur 12. Ersättning av theme_links för att renderera menyträd	26
Figur 13. Ersättning av theme_menu_link för att inkludera eventuella underlänkar	27
Figur 14. Ersättning av funktionen theme_breadcrumb i filen template.php	29
Figur 15. Utskrift av breadcrumbs-elementet i page.tpl.php	29
Figur 16. Basic page inlägg på webbplatsen	30
Figur 17. Klubb/Organ/Utskott inlägg på webbplatsen.....	32
Figur 18. Showcase implementation på Colangines demonstrationsstartsida (Themeforest.net 2014)	33
Figur 19. Klassinställningar för vyformatet	34
Figur 20. Namngivningsalternativ för vyn klubbars mallar	35
Figur 21. Strukturen av ett list-element för showcase.....	36
Figur 22. Mallen views-view-fields--klubbar.tpl.php	36
Figur 23. Listningssida som implementerar showcase-presentation	37
Figur 24. Blogginlägg på webbplatsen	39
Figur 25. Mallen node--blog-entry.tpl.php.....	40
Figur 26. Huvdlistningen för bloggar	41
Figur 27. Framsidans blogglistning	42

Figur 28. Sidofältslistning av bloggar	42
Figur 29. Funktionen build_date_markup.	43
Figur 30. Evenemanginlägg på webbplatsen	45
Figur 31. Inplacering av fält i node_info-variabeln.....	45
Figur 32. Framsidans händelsekalender	46
Figur 33. Uppbyggnad av HTML-stukturen för implementering av prettyPhoto	47
Figur 34. Återanvändning av views-view-fields--klubbar.tpl.php som views-view-fields- -gallery.tpl.php	48
Figur 35. Huvudlistning för galleriinlägg.....	48
Figur 36. Definiering av ny klass i den ersatta mallen forum-topic-list.tpl.php.....	49
Figur 37. Innehållet för mallen views-view-fieds--banner.tpl.php.....	54
Figur 38. Startsidan med bildspel, banner, relevanta inlägg och partners.....	55
Figur 39. Startsidans nedre fält med vyer för bloggar, händelsekalender och gallerier .	56
Figur 40. Aktuellt sidan på webbplatsen	58
Figur 41. Alternativen för lokaliserat innehåll	59
Figur 42. Val av campus.....	61
Figur 43. Framsidan för webbplatsen	62

Förkortningar

URL	Uniform Resource Locator
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart

1 INLEDNING

1.1 Utgångspunkt

Detta arbete är utfört som ett projekt för min nuvarande arbetsgivare Nord Software. Syftet för arbetet är att skapa en ny webbplats, SHS.fi, för Svenska Handelshögskolans Studentkår (SHS). Webbplatsen är avsedd att ersätta den gamla webbplatsen shsweb.fi. Den nya webbplatsen ska ge möjlighet för sidadministratörer att skapa och hantera bloggar, evenemang, materialbanker, gallerier och statiska sidor. Webbplatsen ska också ha ett forum som anonyma användare ska kunna använda och administratörer moderera. Eftersom Svenska Handelshögskolans officiella språk är svenska och engelska och på grund av att skolan har campus i både Helsingfors och Vasa ska webbsidan även vara tvåspråkig och innehållet ska efter behov kunna filtreras efter önskat campus. Webbplatsen är byggd med Drupal 7 som är ett PHP-baserat innehållshanteringssystem.

1.2 Målsättning

Examensarbetets målsättningar är att skapa en tvåspråkig webbplats med Drupal 7 och implementera ett förvalt tema som inte var byggt för Drupal. På det teoretiska planet är målsättningen att ge en grunduppfattning över den struktur som Drupal 7 erbjuder för innehållshanteringen samt att redogöra för hur ett Drupal-tema är uppbyggt och hur ett tema kan modifieras genom att ersätta standardfunktionalitet. För den praktiska delen av arbetet är målet att klargöra hur man utvecklar en innehållsstruktur i Drupal 7 och hur man tillämpar ett existerande tema för presentation av innehållet. Andra mål är att redogöra för de tekniska lösningar som genomförts vid utvecklingen av SHS.fi samt de fallgropar som stöttes på under projektet.

1.3 Metoder

I den teoretiska delen av arbetet byggs först en grunduppfattning om Drupal 7 som innehållshanteringssystem upp med en genomgång av de element som används vid

innehållsstrukturering och Drupal-utveckling. Därefter redogörs för hur ett Drupal-tema är uppbyggt genom forskning i ett temas struktur och modifikationsmöjligheter.

Den praktiska delen klargör utvecklingsprocessen för webbplatsens innehåll och dess presentationslösningar. Redogörelsen av utvecklingsprocessen byggs upp på kunskap från teoridelen.

1.4 Begränsningar

Detta arbete kommer inte att omfatta installationen av Drupal 7. Arbetet kommer inte heller att behandla inloggning, registrering eller annan användarfunktionalitet. Eftersom ett förvalt tema använts för installationen kommer arbetet inte heller att behandla grafiska planering eller temats responsiva aspekter utan främst redogöra för den strukturella implementationen av temat och dess egenskaper.

2 COLANGINE-TEMAT

Colangine - Creative Flat HTML5 Template (Figur 1) är temat som valdes för projektet. Temat är skapat av användaren Theme20 på Themeforest.net där temat även distribuerades (Themeforest.net 2014). Colangine erbjuder en responsiv och pekskärmvänlig design med 66 stycken HTML5- och CSS3-drivna sidmallar för bl.a. portfolio och blogg. Temat utnyttjar JavaScript för presentation av rullgardinsmenyer och har implementerat jQuery paket för stöd av ett flertal bildspelsmoduler, listningsalternativ och mediaspelare. Designen är uppbyggd enligt ett rutnät uppdelat i tolv kolumner vilket stöder inkludering av sidbalkar och enkel strukturering av sidelement. Utöver mallarna och de olika presentationskomponenterna erbjuder även temat en stor samling ikoner, färdiga designer för HTML-element och koder för presentationsalternativ. Temat söktes fram av projektchefen och godkändes av SHS som stilriktning.



Figur 1. Colangine - Creative Flat HTML5 Template produktbeskrivning (Themeforest.net 2014)

Valet av de element av temat som skulle användas var en pågående process där kundrepresentanterna och Nord Softwares representanter höll veckomässiga möten där förverkligandet av designen diskuterades. Argumenten för valen av komponenter att implementera var i första hand ett resultat av hur arbetsdryg en tillämpning av komponenten ansågs vara, dess estetiska värde samt användarvänligheten på administrationssidan av elementet.

Colangine är inte direkt tillämpat för Drupal utan de komponenter som temat erbjuder har implementerats efter behov. Temats färgskala ändrades efter de färgdirektiv som givits av SHS. Rullgardsmenyerna samt den responsiva mobilmenyn implementerades för huvudmenyerna. I sidhuvudets huvudmeny användes även den fastklistrade menyraden vid fullskärmsvisning av sidan. Av bildspelsalternativen valdes endast jQuery-paketet PrettyPhoto för visning av bildalbum och de övriga alternativen ersattes med Drupal-moduler som erbjöd bättre samspel med användargränssnittet. För presentation av inlägg med förknippat bildmaterial användes den framställning som portfoliomallarna erbjöd och för blogginlägg användes de sidmallar som temat erbjöd. Den CSS3 funktionalitet som gjorde att sidkomponenter gled in på hemsidan när sidan laddades skalades senare av eftersom sidans presentation blev långsam och eftersom effekten efterhand ansågs vara en alltför tung del av sidpresentationen.

3 DRUPAL 7

Produkten av detta arbete är byggd på Drupal 7. Drupal är ett innehållshanteringssystem med öppen källkod som ursprungligen skapades 1999 av Dries Buytaert. Av alla sidor på internet byggda på CMS står Drupal för 2,87% (BuiltWith.com 2014). Plattformen är distribuerad under GNU-licens vilket gör att Drupal fritt får användas till vilket som helst ändamål. På grund av detta är Drupal konstant under utveckling och stöder de nyaste webbteknologierna. (Drupal.org 2014a)

Drupal är ett modulärt ramverk och GNU-licenseringen gör att utvecklare fritt kan publicera extra funktionalitet för Drupal i fristående moduler. Detta betyder att då Drupal redan i dess basinstallation har funktionalitet för att bygga avancerade hemsidor har vem som helst möjlighet att ladda ner dessa kontribuerade moduler och på det sättet utvidga funktionaliteten ytterligare. Eftersom allt detta har öppen källkod kräver Drupal relativt lite kodande och projekt går snabbare att färdigtställa.

Plattformen är byggd på programspråket PHP och för visuella element har en hel del JavaScript använts (Melançon et al. 2011, s. xliv). För att spara data används en databas av typen MySQL, PostgreSQL eller SQLite (Drupal.org 2014a).

3.1 Drupals struktur

Drupal är ett innehållshanteringssystem och har byggstenar för de strukturella element som förekommer på hemsidor, en del *core*-moduler inkluderade i basinstallationen och en del genom kontribuerade moduler. Det viktigaste elementet är noder som används som behållare för all typ av innehåll, dessa noder kan kategoriseras med taxonomier. Med den kontribuerade modulen *Views* kan användaren skapa vyer med modifierbara listor av innehåll som sedan kan skrivas ut i block.

3.1.1 Noder

Allt innehåll på en Drupal-sida är representerat av en nod. En nod är en samling av information som bygger upp t.ex. ett blogginlägg eller en artikel. Noder delas upp i innehållstyper för att skilja åt olika typer av innehåll. (Drupal.org 2014b)

Innehållstyperna är fördefinierade samlingar av datatyper som kallas fält. Dessa fält är universella element inom Drupal och ett existerande fält kan återanvändas i samtliga innehållstyper. Rubrikfältet och fältet för brödtext används exempelvis för samtliga innehållstyper som standard. (Drupal.org 2014c)

Nodernas funktionalitet förses av modulen *Node* som ingår i Drupals standardinstallation. Modulen ger möjligheten att skapa, radera och editera noder och innehållstyper samt tillskriva rättigheter för innehållstyperna. (Drupal.org 2014b)

3.1.2 Block

Block är lådor med innehåll som t.ex inloggningsformuläret eller innehållslistor. Dessa block kan placeras i specifika regioner av websidan via Drupals användargränssnitt. Regionerna beskrivs i kapitel 4.2 Temats uppbyggnad. Största delen av blocken byggs oftast upp automatiskt av moduler men kan även skapas efter behov via användargränssnittet. Efter att de skapats kan blockens utseende, innehåll och plats modifieras. Modulen *Block* är inkluderad i standardinstallationen. (Drupal.org 2014d)

3.1.3 Menyer

Menyer i Drupal är samlingar av länkar ämnade för att navigera på en hemsida. Modulen *Menu* är inkluderad i Drupals standardinstallation och ger användaren möjlighet att skapa hierarkiska listor av länkar som sedan kan placeras ut på hemsidan som block. (Drupal.org 2014e)

3.1.4 Taxonomi

Modulen *Taxonomy* ger administratörer möjligheten att klassificera innehållet på Drupal-sidan med hjälp av nyckelord som kallas termer. Termerna definieras för innehållet som exempelvis kategorier eller taggar. (Drupal.org 2014f)

Klassificeringen påbörjas genom att skapa en vokabulär, en lista för en typ av termer. En Drupal-sida kan ha ett obegränsat antal vokabulärer som i sin tur kan ha ett obegränsat antal termer. Termerna inuti en vokabulär kan vara hierarkiskt ordnade så även en term kan ha ett obegränsat antal undertermer. Klassificeringen av innehåll kan vara nyttigt för sidans navigationslogik såväl som arbetsflöde och visningsalternativ. (Drupal.org 2014f)

3.2 Moduler

Moduler är små kodpaket som utökar Drupals funktionalitet. Då Drupal installerats finns redan ett flertal moduler till förfogande och några, som t.ex. modulen *Node*, är aktiverad som standard. (Melançon et al. 2011, s. 14)

Det finns tre typer av moduler. Modulerna som följer med Drupal-installationen kallas *Core*-moduler och moduler som fritt kan laddas ner från Drupal.org kallas kontribuerade moduler. Den tredje typen kallas anpassade moduler och dessa är skapade specifikt för ett projekt och är av ingen nytta för andra projekt. (Drupal.org 2014g)

Kontribuerade moduler är till stor nytta för ett Drupalprojekt eftersom i princip vilken funktionalitet som helst kan ha en färdig modul för ändamålet. Vissa kontribuerade moduler som t.ex. *View* anses som ett måste för vilket Drupal-projekt som helst. Eftersom vem som helst kan publicera en Drupal-modul lönar det sig att kolla upp modulens info-sida på Drupal.org för att kolla efter allmänna klagomål och buggar.

4 TEMAUTVECKLING I DRUPAL 7

Då Drupal 7 har installerats använder installationen temat Bartik som standard. Bartik temat är mycket dynamiskt och fungerar i samspel med Drupals kärnmoduler och även de mest allmänna kontribuerade modulerna.

För att göra en Drupal-sida mer personlig byggs i de flesta fall ett nytt tema, ibland baserat på ett existerande Drupal-tema för att spara tid. I så fall skapas ett undertema som ersätter existerande element i det ursprungliga temat efter behov. I fallet av detta arbete skiljer sig strukturen av Colangine-temat från Drupals inkluderade teman till den grad att det nya temat lönar sig att byggas upp från grunden enligt den funktionalitet som sidan kräver och Colangine-temat erbjuder. Eftersom Colangine-temat inte ursprungligen är byggt för Drupal behöver således inget undertema skapas.

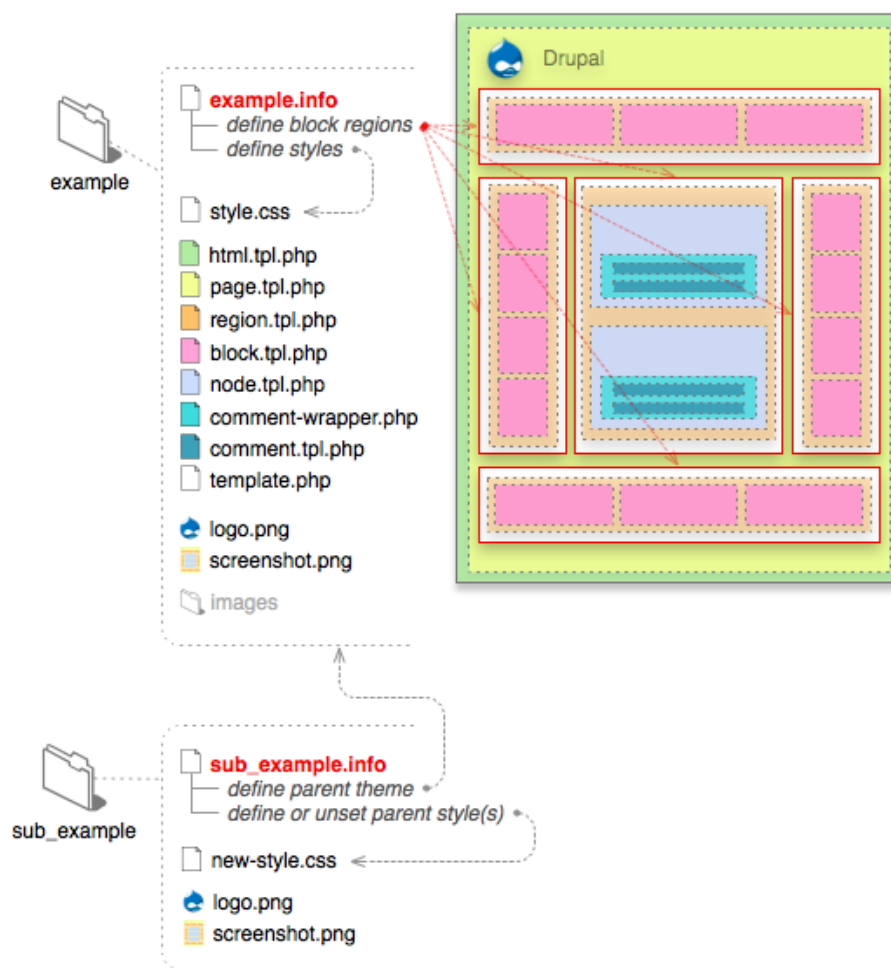
4.1 Vad är ett tema

Ett tema i Drupal är en samling filer som visuellt utformar det innehåll som Drupal och dess grundläggande och utvidgande moduler producerar. Samlingen består främst av filer av typen PHP, CSS och möjligen Javascript. (Aschan 2011, s. 13)

Konceptuellt är temat en behållare som sköter om formateringen och visningen av data på skärmen. Filerna i behållaren är av den typ som grafiska webbdesigners använder på dagligt basis d.v.s. formatmallar, märkspråk, bilder och skript. Utöver de vanliga filtyperna används även *.tpl.php* som utnyttjas av Drupals temamotor *PHPTemplate*. (Shreves 2011, s. 11-12)

4.2 Temats uppbyggnad

Layouten i Drupals teman är uppbyggd av s.k. regioner där användaren sedan via Drupals administrativa gränssnitt kan placera block (Figur 2). Regionerna definieras i filen *.info* och kan efter det skrivas ut i mallar. I filen *.info* definieras också de CSS- och JavaScript filer som ansvarar för sidans estetiska element. Denna fil är det enda som krävs för att temat ska registreras av Drupal-installationen. (Drupal.org 2014h)

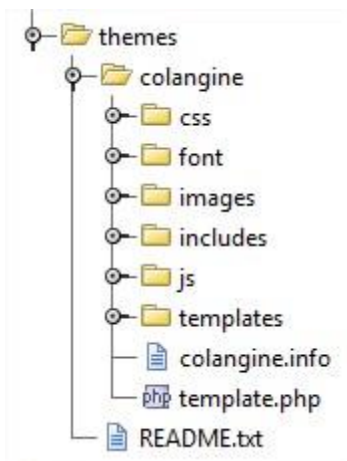


Figur 2. Beskrivning av strukturen i ett Drupal 7 tema (Drupal.org 2014a)

Sidstrukturen i ett tema manipuleras av mallar som existerar i temakatalogen. Dessa filer fungerar som mallar för sid- och komponentstrukturer inom Drupal och är uppbyggda med HTML som märkspråk och använder sig av PHP variabler för innehållet. Mallarna är inte obligatoriska för temat eftersom alla nödvändiga filer redan finns i Drupals basinstallation. De mallar som läggs till i temakatalogen ersätter de existerande mallarna med samma namn i Drupal-systemet. (Drupal.org 2014h)

För övrig kod som behövs för temats logik kan utvecklaren skapa en fil med namnet *template.php*. I denna fil placeras de funktioner som ersätter standardfunktioner i Drupal vars uppgift är att formatera den data som ska skrivas ut i mallarna. Filen *template.php* är inte heller obligatorisk för ett Drupal-tema eftersom all standardfunktionalitet redan finns tillämpad. (Drupal.org 2014h)

Även om Drupal hittar samtliga filer direkt i temakatalogens rot kan katalogen lätt bli rörig på grund av antalet filer som inkluderas i temat. För överskådlighet skapas oftast underkataloger för CSS-, JavaScript- och mallar (Figur 3). Även bildmaterial och andra typer av filer bör struktureras i egna underkataloger. (Rambeck 2014)



Figur 3. Temakatalog för Colangine temat

4.2.1 Filen .info

För att Drupal ska registrera temat behövs endast konfigurationsfilen *.info*. Filnamnet bestäms av temats namn följt av filändelsen ".info". För temat Colangine blir det fullständiga filnamnet *colangine.info*. Filen är det enda som krävs för att Drupal ska registrera temat och är även den enda obligatoriska filen i ett Drupal 7 tema.

Filen är en statisk textfil med information i form av nyckelvärden. I denna fil behöver skaparen endast uppge temats namn och versionen på den Drupal-installation temat byggts på för att *.info*-filen ska anses giltig av Drupal. Utöver dessa nyckelvärden är det i denna fil som temats regioner definieras och där utomstående script- och stilfiler kan inkluderas i temats HTML *<head>*-tag (Figur 4). Alla ändringar till *.info*-filen kräver att Drupal's cache töms för att de ska träda i kraft. (Drupal.org 2014i)


```

name = Colangine
description = Drupal implementation of Colangine for SHS.fi
package = SHS
version = 0.1
core = 7.x

stylesheets[all][] = css/style.css
stylesheets[all][] = css/custom.css
stylesheets[all][] = css/font-awesome.css
stylesheets[all][] = css/custom-showcase.css

scripts[] = js/jquery.jcarousel.min.js
scripts[] = js/rs-plugin/plugin/sources/jquery.themepunch.plugins.min.js
scripts[] = js/rs-plugin/js/jquery.themepunch.revolution.min.js
scripts[] = js/superfish.js
scripts[] = js/jquery.prettyPhoto.js
scripts[] = js/jquery.easing.1.3.js
scripts[] = js/jquery.flexslider-min.js
scripts[] = js/twitter/jquery.tweet.js
scripts[] = js/custom.js
scripts[] = js/modernizr.custom.28468.js
scripts[] = js/modernizr.custom.46884.js

regions[header] = Header
regions[options] = Options
regions[page_top] = Page top
regions[hero] = Hero
regions[frontpage_banner] = Frontpage banner
regions[frontpage_features_1] = Frontpage Features 1
regions[frontpage_features_2] = Frontpage Features 2
regions[page_bottom] = Page bottom
regions[highlighted] = Highlighted

regions[list_page_feature] = List page feature
regions[list_page_left] = List page left
regions[list_page_right] = List page right

regions[featured] = Featured
regions[content] = Content
regions[sidebar_first] = Sidebar first
regions[sidebar_second] = Sidebar second

regions[footer_left] = Footer left
regions[footer_center] = Footer center
regions[footer_right] = Footer right

```

Figur 4. colangine.info-filen för Colangine- temat

Då man implementerar ett existerande HTML tema för Drupal erbjuder *.info*-filen en effektiv lösning för att inkludera de CSS och JavaScript filer som temat kräver. De filer som inkluderats via *.info*-filen blir inkluderade på samtliga sidor i temat så det lönar sig att överväga om effektiviteten kan ökas genom att inkludera de mindre använda filerna dynamiskt i ett senare skede (Melançon et al. 2011, s. 343-344). Det är möjligt att använda sig av de regioner som används i standardinstallationen och användningen av dem gör även strukturen enklare att tyda för utvecklare som är bekanta med Drupal. Namngivningen av regionerna är dock fri.

4.2.2 Mallar

För att en sida ska få en struktur i HTML-format måste temats visuella element använda sig av mallar i form av *tpl.php-filer*. Alla element i Drupal har egna mallar i Drupals standardinstallation. Dessa mallar kan ersättas genom att skapa en fil i temats katalog med samma filnamn som standardmallen i Drupals system och sedan tömma Drupals cache. Valet av mall sköts av *PHPTemplate* temamotorn som används inom Drupal. Nya unika mallar för sidkomponenter, som inte redan existerar i Drupals kärna, kan också definieras efter behov. Filerna är av typen PHP och innehåller HTML med sekvenser av PHP (Figur 5). Sekvenserna används för att skriva ut PHP-variabler som står för sidans dynamiska innehåll. (Melançon et al. 2011, s. 293-296)

```
<div class="breadcrumb">
  <div class="row clearfix">
    <ul class="breadcrumbs">
      <?php print $breadcrumb; ?>
      <li>
        <h3><?php print $title; ?></h3>
      </li>
    </ul>
  </div>
  <!-- row -->
</div>
<!-- breadcrumb -->
```

Figur 5. HTML-märkspråk med PHP-sekvenser

Om sidan som ska omstruktureras använder en mall som annat material är beroende av kan mallen ges ett mer specifikt filnamn. Om t.ex framsidan kräver en annan sidstruktur än de övriga sidorna på hemsidan kan mallen ges filnamnet *page--front.tpl.php*. Temamotorn kommer i fortsättningen vid renderingen av framsidan att ignorera *page.tpl.php*-filen och i stället använda den mer specifika mallen *page--front.tpl.php*. Övriga sidor kommer dock fortfarande använda *page.tpl.php* (Melançon et al. 2011, s. 305-307)

Då det är frågan om att implementera ett existerande HTML-tema är det främst mallarna man kommer att jobba med. Den existerande HTML-strukturen bör delas upp i mindre helheter i form av mallar för att senare dynamiskt kunna utnyttja dem under resten av projektet. För att

tillämpa strukturen för Drupal skalas all konkret information (brödtext och bilder) bort och ersätts med dynamiska värden i form av PHP-variabler (Figur 6).

(Drupal.org 2014j)

```
<?php
/** @file ... */
?><!DOCTYPE html>
<!--[if lt IE 7]><html class="ie ie6" lang="en"> <![endif]-->
<!--[if IE 7]><html class="ie ie7" lang="en"> <![endif]-->
<!--[if IE 8]><html class="ie ie8" lang="en"> <![endif]-->
<!--[if (gte IE 9)!!(IE)]><!-->
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php print $language->language; ?>" version="XHTML+RDFa 1.0"
  dir="<?php print $language->dir; ?>"<?php print $rdf_namespaces; ?>>

<head profile="<?php print $grddl_profile; ?>">
  <?php print $head; ?>
  <title><?php print $head_title; ?></title>
  <?php print $styles; ?>
  <link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Oswald">
  <?php print $scripts; ?>

  <!--[if IE]>
  <meta http-equiv="X-UA-Compatible" content="IE=9; IE=8; IE=EmulateIE8; IE=EDGE"/>
  <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
  <!--[if IE 8]>
  <link rel="stylesheet" type="text/css" href="css/ie.css" media="screen"/><![endif]-->
  <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1">
</head>
<body class="<?php print $classes; ?>"><?php print $attributes;?>>
<div id="skip-link">
  <a href="#main-content" class="element-invisible element-focusable"><?php print t('Skip to main content'); ?></a>
</div>
<?php print $page_top; ?>
<?php print $page; ?>
<?php print $page_bottom; ?>
</body>
</html>
```

Figur 6. Drupal-temat Colangines *html.tpl.php*-fil byggd med HTML5-temat Colangine som grund

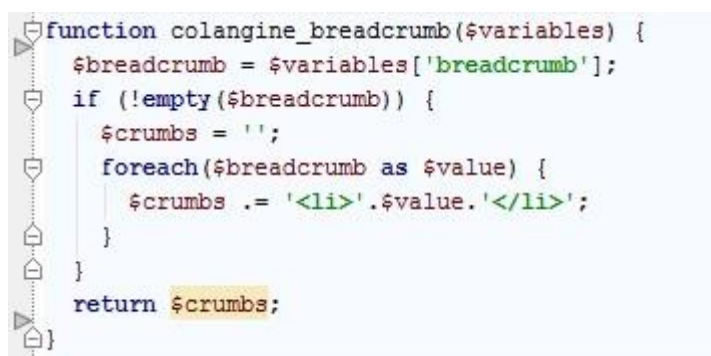
4.2.3 Ersättning av Drupals temafunktioner

Utöver sidmallarna struktureras även sidelement av temafunktioner. Dessa funktioner skapar dynamiskt HTML-kod på samma sätt som en mall men gör det i ren PHP form. Strukturen byggs upp i strängar av HTML konkatenerade med PHP variabler, stängarna konkateneras efterhand till en slutlig utskriftssträng som sedan returneras i slutet av funktionen i fråga.

En temafunktion kan ersättas genom att skapa en ny funktion i temats *template.php*-fil. För att funktionen ska hittas måste den namnges enligt modellen *temats namn_funktionens namn*. Drupals standardfunktioner använder namnet "theme" och för att funktionens ska hittas ersätts "theme" av det nya temats namn. Exempelvis skulle funktionen *colangine_button* i

Colangine-temats *template.php*-fil ersätta standardfunktionen *theme_button*. (Melançon et al. 2011, s. 308-309)

Temafunktioner används främst inom temautveckling då det är fråga om inbyggda Drupal-komponenter som ska modifieras för att passa in i det nya temat (Figur 7). Vanliga sidelement som kan behöva ersätta temafunktioner är olika former av länkar som t.ex länkstigen och ”Visa mer”-länkar. Ersättning av dessa temafunktioner gör även temat mer användarvänligt vid vidaredistribution av temat.

A screenshot of a code editor showing a PHP function named `colangine_breadcrumb`. The function takes `$variables` as an argument. It assigns `$variables['breadcrumb']` to `$breadcrumb`. It then checks if `$breadcrumb` is not empty. If it is not empty, it initializes `$crumbs` as an empty string and enters a `foreach` loop over `$breadcrumb` as `$value`. Inside the loop, it appends ``, the value, and `` to `$crumbs`. After the loop, it returns `$crumbs`.

```
function colangine_breadcrumb($variables) {  
    $breadcrumb = $variables['breadcrumb'];  
    if (!empty($breadcrumb)) {  
        $crumbs = '';  
        foreach($breadcrumb as $value) {  
            $crumbs .= '<li>'.$value.'</li>';  
        }  
    }  
    return $crumbs;  
}
```

Figur 7. Ersättning av temafunktionen *theme_breadcrumb*

4.2.4 Förprocessering

Mallarna i Drupal 7 får information i form av variabler som hämtats från Drupals databas och processerats i Drupals standardfunktioner (Drupal.org 2014k). Via Drupals användargränssnitt kan användare ändra på bl.a. namn på klasser i HTML-strukturen och även sidinnehåll kan formateras i viss mån via gränssnittet. Förr eller senare kommer man dock behöva formatera information på ett sätt som användargränssnittet inte ger möjlighet till och då blir förprocessering blir aktuell.

I en förprocesseringsfunktion kommer man åt att ändra på variabler före de skickas till en mall. Det betyder att existerande variabelvärden kan ersättas med nya värden eller möjligen raderas. Även helt nya variabler kan definieras och på så sätt skickas till mallarna.

En förprocesseringsfunktion placeras, på samma sätt som temafunktionerna, i *template.php*-filen (Figur 8). Namngivningen fungerar enligt modellen *namnet på*

temat_preprocess_komponentt som ska förprocesseras. Funktionen *colangine_preprocess_page* processerar således variablerna som skickas till mallar som implementerar någon variation av *page*-mallen. Förprocesseringsfunktionerna kräver att *variables* parametern skickas till funktionen som referens eftersom funktionerna ändrar på listan av variabler direkt och i sig inte returnerar något. (Shreves 2011, s. 106)

```
function colangine_preprocess_comment(&$variables){
  unset($variables['classes_array']);
  $variables['classes_array'][] = 'user-comments';
  $variables['classes_array'][] = 'mbs';

  $variables['comment_author'] = $variables['user']->name;
  $variables['comment_created'] = date('d.m.Y', $variables['comment']->created);
}
```

Figur 8. Förprocesseringsfunktion för kommentarer

4.2.5 CSS- och Javascript-filer

Det finns två metoder att implementera CSS- och JavaScript i ett tema. Den enklare metoden är att definiera filerna i *.info*-filen vilket laddar dessa filer på samtliga sidor. (Melançon et al. 2011, s. 343)

Den andra metoden är att i t.ex. *template.php*-filens förprocesseringsfunktioner lägga till CSS-filer med funktionen *drupal_add_css* och JavaScript-filer med funktionen *drupal_add_js*. Med funktionerna ger man möjligheten åt Drupal att lägga till kod eller hela filer efter behov. Denna metod är att föredra då det är fråga om kod som endast används på ett fåtal sidor då andra sidor där filerna inte används kommer att laddas snabbare. (Melançon et al. 2011, s. 343-344)

Eftesom ett tema oftast har ett stort antal CSS-filer att ladda på en sida är det skäl att komprimera filerna då hemsidan läggs upp på produktionsservern. Drupal 7 har en inbyggd funktion i användargränssnittet där man kan aggregera och komprimera filerna till en enda fil där mellanslag och radbyten skalats bort. Detta resulterar i en mindre fil med lägre laddningstid. (Shreves 2011, s. 87)

På samma sätt som ett tema kan ersätta standardfunktionalitet ersätter det nya temats CSS de tillämpningar som definieras på nytt. (Shreves 2011, s.88)

4.2.6 Temarelaterat bildmaterial

Som andra teman har även teman inom Drupal ofta bildmaterial inkluderat antingen direkt i HTML-strukturen eller inkluderade genom klasser via CSS. Var bilderna placeras spelar ingen större roll så länge temat klarar av att hitta dem i filträdet för hemsidan. För överskådliggetskull är det dock god praxis, då bildmaterialet är rikligt, att skapa en "images" katalog där alla temarelaterade bilder sparas (Duckett 2011, s. 98).

Bilder inkluderade via CSS-egenskaper, t.ex. *background-image*, följer samma mönster som exklusiva HTML sidor. I Drupal-temats katalogträd betyder detta att sökvägen för bildmaterialet i CSS-filer blir "../images/bild.jpg" då CSS-filerna och bilderna har placerats i egna kataloger (Kumar 2012, s. 84). Då man använder ett färdigt HTML tema stöter man sannolikt på bilder inkluderade via HTML taggen **. Eftersom mallarna används dynamiskt inom Drupal kan den URL som associeras till mallen variera kraftigt och sökvägen måste därför vara absolut. Lösningen till detta är att använda PHP-variabeln *\$directory* som returnerar den aktiva sökvägen (Drupal.org 2014). Koden för att skriva ut en statisk bild i en mall blir således *?php print \$directory; ?>/images/bild.jpg" alt="Bild">*.

5 UTVECKLING AV SHS.FI

Överföringen av Colangine-temat samt den funktionalitet som shsweb.fi erbjudit till Drupal-ramverket var en simultan process. Detta eftersom Colangine-temat krävde sidinnehåll för att kunna implementeras och sidinnehållets struktur i sin tur gynnades av att ha temat som riktlinje.

För att redan från början hålla Colangine som huvudinriktning gjordes en grundimplementation av HTML5-temat för Drupal-installationen där de viktigaste funktionerna som visningen av framsidan, menyerna och presentationeskomponenterna tillämpades. Efter grundimplementationen användes Drupals användargränssnitt för att skapa en innehållstyp vars visning sedan modifierades på basis av Colangine-temats mallar och

erbjudna presentationsalternativ. Efter att strukturen och presentationen för en innehållstyp skapats byggdes listningsalternativen för den ifrågavarande innehållstypen. På detta sätt kunde tidigare tillämpningar användas igen eller möjligen modifieras för att stöda följande innehållstyp. Efter att listningsalternativen byggts upp var det möjligt att använda dem på listningssidor som framsidan och aktuellt sidan.

5.1 Grundimplementation av Colangine-temat

För att börja implementeringen skapades temakatalogen *colangine* i Drupals katalog *themes* och inuti *colangine*-katalogen skapades filen *colangine.info* med det basdata som krävdes för att temat ska registreras. I den nyskapade katalogen kopierades samtliga kataloger ur den temakatalog som *ThemeForest.com* förser vid köpet av Colangine-temat. Temats *style.css*-fil flyttades in i en egen katalog och inkluderades sedan i *colangine.info*, även alla JavaScript-filer i katalogen *js* inkluderades i *.info*-filen. Därefter tömdes Drupals cache och temat kunde aktiveras i Drupals administrationsgränssnitt. Filerna valdes att inkluderas direkt i *.info*-filen eftersom temat endast erbjuder en CSS-fil för hela temat. Den specifika användningen av JavaScript-filerna var i begynnelseskedet oklart så samtliga filer inkluderades i *.info*-filen.

Efter att temat aktiverats som standard för Drupal-installationen skapades mallar för att återskapa Colangine-temats *index.html*-fil som en statisk fungerande sida. Detta för att se till att alla filer och skript som behövdes fanns tillgängliga och fungerade som de skulle. För implementeringen ersattes standardmallen *html.tpl.php* och mallen *page--front.tpl.php* skapades som ersättare av *page.tpl.php* på huvudsidan.

Först ersattes standardmallen *html.tpl.php*. Innehållet för filen skapades genom att kopiera innehållet ur filen *index.html* i Colangine-temat. Eftersom *html.tpl.php* är ämnad att endast stå för HTML-filens huvud raderades all information inom *<body>* taggen och ersattes med PHP-variabler för att skriva ut innehållet på sidan. Den initierande *<body>* taggen modifierades även till att motsvara den ursprungliga *html.tpl.php* filens tagg (Figur 9).

```

<body class="<?php print $classes; ?>" <?php print $attributes;?>>
  <div id="skip-link">
    <a href="#main-content" class="element-invisible element-focusable">
      <?php print t('Skip to main content'); ?>
    </a>
  </div>
  <?php print $page_top; ?>
  <?php print $page; ?>
  <?php print $page_bottom; ?>
</body>

```

Figur 9. Det modifierade `<body>`-elementet

Den ursprungliga `<head>`-taggen modifierades till att skriva ut de baselement som Drupal genererar för individuella sidor. I praktiken betydde detta att *title*-elementet ersattes med variabler och `<link>`-taggarna som inkluderade CSS- och JavaScript-filer ersattes med motsvarande variabler. I övrigt skedde ändringarna i filen för att få den att överensstämma mer med den ursprungliga *html.tpl.php*-filen.

Filen *page--front.tpl.php* strukturerar det innehåll som skrivs ut i *html.tpl.php*-filens `<body>`-tagg. I början bestod *page--front.tpl.php* av Colangine-temats *index.html*-fils *body* innehåll. Bildmaterialet som fanns inkluderat i *index.html* kunde inte använda sig av relativa länkar till temakatalogens bildmaterial så dessa länkar gjordes dynamiska för att förse testsidan med det bildmaterial som krävdes. Efter att ha testat att all funktionalitet fanns tillgänglig var Colangine-temat klart att börja utvecklas för Drupal.

5.1.1 Sidhuvud och sidfot

Eftersom sidhuvudet och sidfoten oftast hålls den samma på alla sidor kan ett flertal *page.tpl.php*-mallar skapa onödigt arbete då uppdateringar på dessa element ska ändras. För att undvika denna möjliga arbetsbörda skapades filerna *header.php* och *footer.php* som innehöll respektive komponents struktur i form av PHP-funktioner. En *init.php*-fil skapades för att ackommodera inkluderingarna av dessa filer i *template.php*-filen varefter funktionerna *colangine_header* och *colangine_footer* kunde användas för att skriva ut elementen på önskad plats i *page.tpl.php*-mallarna.

I sidhuvudet ersattes de statiska menyerna med drupals egna menyer och *page*-mallens variabler för sidans logotyp tillämpades för att anknyta användargränssnittet till logotypens modifikation. Sökfältet ersattes med en ny region för val av språk och campus. Sidfoten delades upp i tre regioner där block för sidinformation kunde placeras.

5.1.2 Implementering av rullgardinsmeny

Efter skapandet av en menystruktur med Drupals inkluderade *Menu*-modul och utskriften av den i sidhuvudet med funktionen *theme* (Figur 10) stod det klart att Colangine-temats rullgardinsmeny inte kunde implementeras. Orsaken var att Drupals inbyggda sätt att skriva ut menyer endast skriver ut den första nivån av en menys hierarkiska trädstruktur. Ett annat problem låg i att Drupal-menyer skrivs ut som listor vars ``-taggs klass inte kan påverkas genom Drupals användargränssnitt. På grund av detta kunde inte Colangine-temats klass tillskrivas menyn och den CSS och JavaScript som byggde upp menyns utseende aktiverades aldrig för menyn.

```
<?php print theme('links_system_main_menu', array(
  'links' => menu_navigation_links('main-menu'),
  'attributes' => array(
    'id' => 'main-menu',
  )
)); ?>
```

Figur 10. Utskrift av meny med funktionen *theme*

Drupal har ett antal tilläggsmoduler som gör det möjligt att använda rullgardinsmenyer. I fallet av Colangine-temat fanns all funktionalitet i form av CSS och Javascript men HTML-strukturen som dessa krävde saknades. De moduler som fanns som alternativ hade den praktiska funktionalitet som söktes men inte det utseende och visuella funktionalitet som eftersträvades. Lösningen blev därför att ersätta temafunktioner för att rätta till strukturen.

För att ge ``-taggen den saknade menyklassen *sf-menu* krävdes en specificerad ersättning av temafunktionen *theme_menu_tree*. Funktionen returnerar utskriftssträngen för en meny och den saknade klassen kan där specificeras. Eftersom det endast är huvudmenyns klass som behövde specificerades funktionens namn till *colangine_menu_tree__main_menu* (Figur 11).

```
function colangine_menu_tree__main_menu($variables) {
  $output = '<ul class="sf-menu">' . $variables['tree'] . '</ul>';
  return $output;
}
```

Figur 11. Ersättning av funktionen *theme_menu_tree* för att ge ``-taggen den saknade klassen *sf-menu*

För att skriva ut menystrukturen i sin helhet ersattes temafunktionen *theme_links* (Figur 12) med *colangine_links*. Detta för att kunna returnera en renderad meny med funktionen *drupal_render* om länkarna i frågan tillhör huvudmenyn i stället för att renderera en länk komponent i taget.

```
function colangine_links($variables) {
  if (array_key_exists('id', $variables['attributes']))
    && $variables['attributes']['id'] == 'main-menu'
  {
    $pid = variable_get('menu_main_links_source', 'main-menu');
    $tree = menu_tree($pid);
    return drupal_render($tree);
  }
  return theme_links($variables);
}
```

Figur 12. Ersättning av *theme_links* för att renderera menyträd

Funktionen *drupal_render* tar emot en lista av variabler och returnerar en HTML-sträng som kan skrivas ut. Bland variablerna hittas *#theme*-variabeln som bestämmer temafunktionen för, i det här fallet, enskilda menylänkar. Temafunktionen i fallet av huvudmenyns länkar är *menu_link__main_menu*. Genom att ersätta basfunktionen *theme_menu_link* med *colangine_menu_link* implementerades möjligheten till rullgardinsmenyer för samtliga menyer på Drupal-sidan. Genom att leta igenom länkelementets variabler i den nya funktionen kan det fastställas om menylänken har underlänkar i menystrukturen. I fallet av att länken har underlänkar placeras dessa i en ny HTML-lista efter länken i det nuvarande listelementet. Funktionen kan även ta reda på om användaren befinner sig på länken som processeras och lägga till en klass för att markera detta (Figur 13).

```

function colangine_menu_link($variables) {
  global $language;
  unset($variables['element']['#attributes']['class']);
  $element = $variables['element'];

  if (isset($element['#localized_options']['attributes']['class'][0])
    && $element['#localized_options']['attributes']['class'][0] == 'active-trail') {
    $element['#attributes']['class'][] = 'current';
  }

  if ($element['#below']) {
    unset($element['#below']['#theme_wrappers']);
    $sub_menu = '<ul>' . drupal_render($element['#below']) . '</ul>';
  } else {
    $sub_menu = '';
  }

  $title = _t('Menu link title', $element['#original_link'], $language->language);
  $output = l($title, $element['#href'], $element['#localized_options']);
  return '<li>' . drupal_attributes($element['#attributes']) . '>' . $output . $sub_menu . '</li>';
}

```

Figur 13. Ersättning av `theme_menu_link` för att inkludera eventuella underlänkar

Eftersom de ändringar som gjorts placerats i temafunktioner kommer inga förändringar implementeras innan Drupals cache tömts och de nya temafunktionerna tagits i bruk.

5.2 Innehållsstruktur

Innehållstrukturen syftar på samlingen av innehållstyper och deras presentation på Drupal-sidan. Varje typ av innehåll på SHS.fi har en vy för presentation av ett enskilt inlägg samt minst en listningsdisplay för presentation och åtkomst av innehållet.

Vid skapandet av en innehållstyp planerades först vilka fält typen krävde för önskad funktionalitet och användarvänlighet. Efter att innehållstypens struktur skapats publicerades ett antal testinlägg för tillämpa Colangine-temats design för typens enskilda inläggs presentationssida. Efter att inläggssidan temats byggdes en eller flera listningsvyer upp med hjälp av *View*-modulen. Efter att listningen konfigurerats att returnera önskat resultat via användargränssnittet modifierades visningen av resultaten så de överstämde med Colangines mallar.

5.2.1 Basic page

Basic page är till för grundinnehållet på SHS.fi. Alla statiska sidor med innehåll som inte gör dem nödvändiga att listas upp på någon annan plats än i en meny använder denna innehållstyp. Eftersom innehållstypen är så grundläggande kräver den inga typspecifika specialfält. Fälten för innehållstypen är följande:

- Title (Text)
- Body (Lång text och sammandrag)
- Image (Bild)
- Fri text (Lång text)
- Relaterade sidor (Entitetsreferens)

Title och *body*-fälten skapas automatiskt då en ny innehållstyp sparas. De är huvudstrukturen för samtliga innehållstyper och titeln är alltid ett obligatoriskt fält vid skapandet av inlägg. Image fältet är till för att ge en visuell representation av ett inlägg i form av en bild i fall inlägget skulle inkluderas i en listningsvy. Fri text och relaterade sidor är fält där administratörer fritt kan inkludera sekundär information i HTML-format eller länka till andra inlägg.

Eftersom innehållstypen är den grundläggande modellen för siduppläggningsen på Drupal-sidan ersattes grundmallarna *page.tpl.php* och *node.tpl.php*. Eftersom en *page*-mall för Colangine, *page--front.tpl.php*, redan hade byggts i grundimplementeringen kopierades mallen och namngavs som *page.tpl.php*. Den kodstruktur som använts för att bygga upp innehållet mellan sidhuvudet och sidfoten byttes ut och ersattes med strukturen för den sidlayout som bestämts för innehållssidorna. Sidans struktur formaterades enligt Colangine-mallen för en sida med ett sidofält till vänster om innehållet. För utskrift av dynamiska element som sidans innehåll, innehållsets administrationsgränssnitt och länkstig konsulterades standardmallen där önskade element kopierades och implementerades. Drupals standard region, *sidebar_first*, valdes dessutom att renderas inuti sidfältet för att block skulle kunna placeras ut efter behov.

Eftersom administrationsgränssnittets utseende var godtyckligt då endast administratörer skulle komma åt att se det behövde den inte modifieras vidare för *page*-mallen. Däremot

skrevs elementet för länkstigen ut på ett sätt som inte motsvarade den struktur som Colangine temats tillämpning krävde. Elementet står för utskriften av navigationsvägen till sidan i fråga. För att ändra på strukturen ersattes temafunktionen som stod för formateringen av *breadcrumbs*-variabeln med en ny funktion i *template.php*-filen. Funktionen utelämnar *breadcrumb*-variabelns ``-tagg (Figur 14) för att tillåta *page*-mallen att själv skriva ut den och lägga till den i frågavarande sidans titel som det sista *list*-elementet (Figur 15).

```
function colangine_breadcrumb($variables) {  
  
    $breadcrumb = $variables['breadcrumb'];  
    if (!empty($breadcrumb)) {  
        $crumbs = '';  
        foreach($breadcrumb as $value) {  
            $crumbs .= '<li>'. $value. '</li>';  
        }  
    }  
    return $crumbs;  
}
```

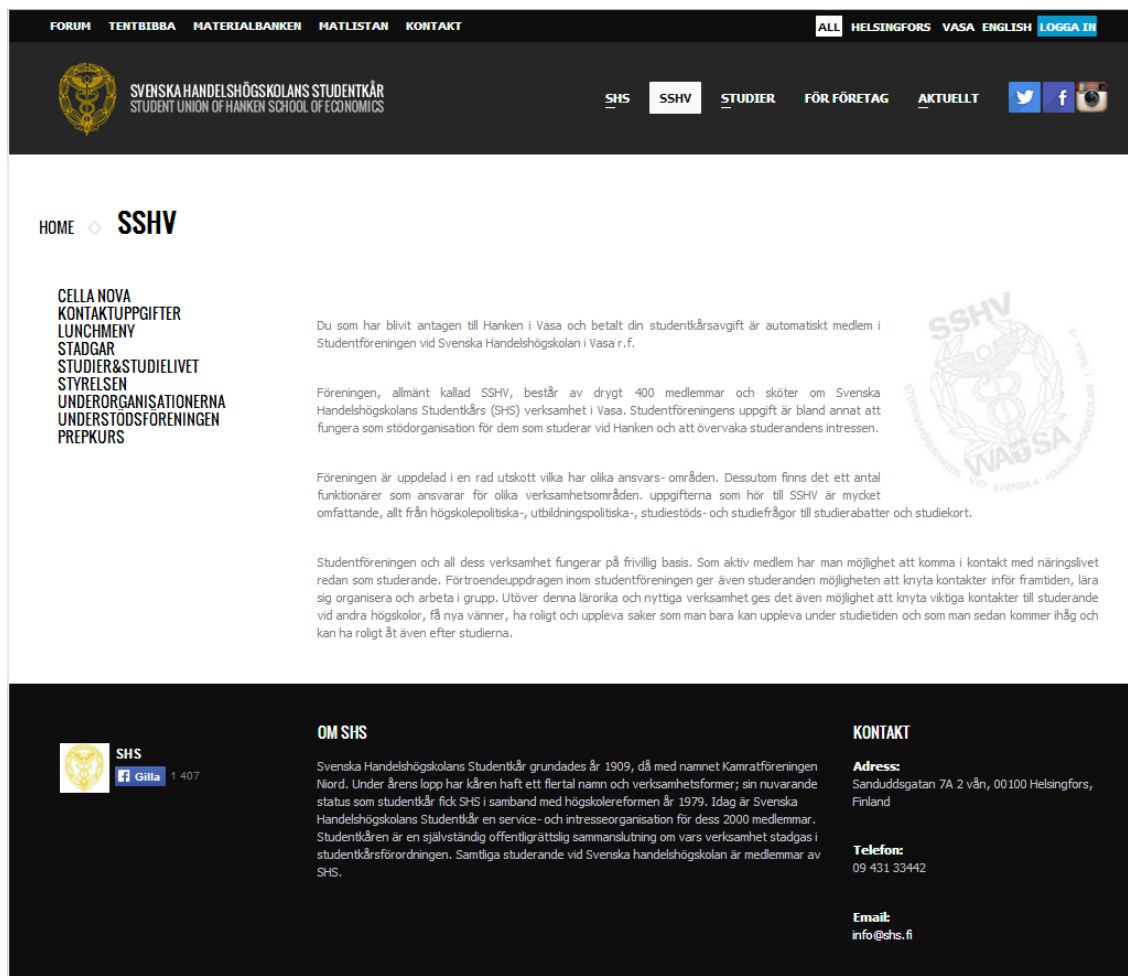
Figur 14. Ersättning av funktionen *theme_breadcrumb* i filen *template.php*

```
<div class="breadcrumb">  
    <div class="row clearfix">  
        <ul class="breadcrumbs">  
            <?php print $breadcrumb; ?>  
            <li><h3><?php print $title; ?></h3></li>  
        </ul>  
    </div><!-- row -->  
</div><!-- breadcrumb -->
```

Figur 15. Utskrift av *breadcrumbs*-elementet i *page.tpl.php*

I den ersättande *node*-mallen gjordes inga andra ändringar än att alla fält förutom *body*-texten och bilden exkluderades vid utskrift. Dessa exkluderade fält konfigurerades så att de i stället skrivs ut under *sidebar_first*-regionen i sidofältet. Detta gjordes genom att i *template.php*-filen skapa en förprocesseringsfunktion för *page*-mallen där valda fält renderades i en ny variabel, *node_info*, som passerades till *page*-mallen. I funktionen placeras *Fri text* och *Relaterade sidor* fälten automatiskt in i *node_info*-variabeln ifall dessa är tillgängliga.

I *sidebar_first*-regionen placerades en ny implementation av huvudmenyn. Där listas de länkar som befinner sig på samma hierarkiska nivå som den ifågavarande sidan samt en undernivå av dessa länkar.



Figur 16. Basic page inlägg på webbplatsen

5.2.2 Klubbar, organ och utskott

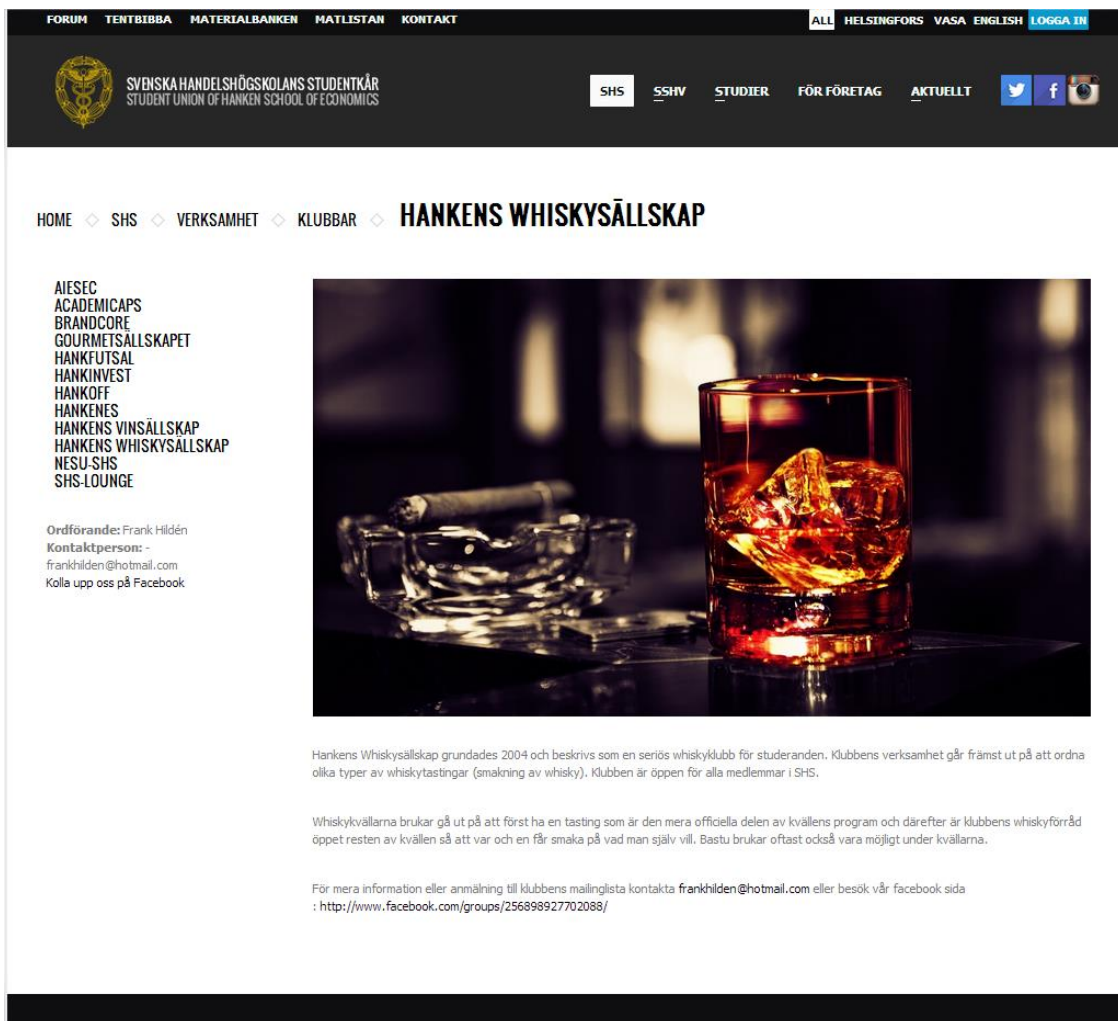
Syftet med innehållstypen är att ge administratörer möjligheten att registrera kontaktsidor för klubbar organ och utskott. Innehållstypen omfattar alla dessa typer av inlägg eftersom den information som bör föras är den samma oberoende vilken typ av organisation det är fråga om.

Innehållstypen skapades ursprungligen för organisationstypen *Klubbar* och innehållet är främst av typen kontaktinformation. Innehållstypen har därför ett flertal fält specifika.

Innehållstypen har följande fält:

- Title (Text)
- Typ (Lista (text))
- Image (Bild)
- Body (Lång text och sammandrag)
- Syfte (Lång text)
- Ordförande (Text)
- Adress (Text)
- Kontaktperson (Text)
- Email (Text)
- Facebook (Link)

Innehållssidan (Figur 17) är ordnad så att innehållsbilden visas högst upp och *Body*-texten därefter. Ordförande, kontaktperson, e-post adressen och länken till organisationens Facebook-sida placeras in i *node_info* variabeln för bättre överskådlighet.

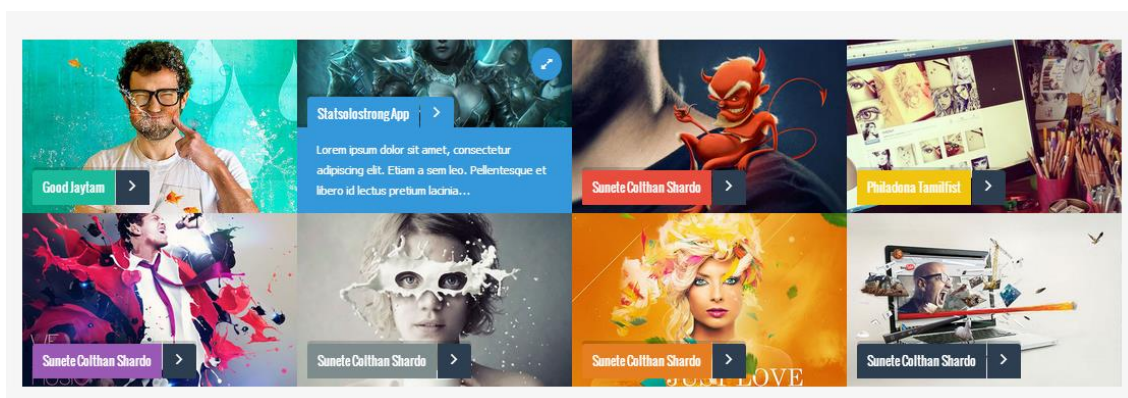


Figur 17. Klubb/Organ/Utskott inlägg på webbplatsen

På listningssidan *klubbar* visas samtliga inlägg med innehållstypen Klubb/Organ/Utskott och som har klubb som värde i Typ-fältet. Sidan har åstadkommit med tillägsmodulen *Views*. Vyn i fråga är en sida som skriver ut en trimmad version av body-fältet, klubbens bild och namnet på organisationen för varje existerande klubb. Resultatet begränsas ytterligare med att inlägget måste vara publicerat.

Eftersom klubbar, organ och utskott alla delar samma fält kan samma vy användas för samtliga typer inom innehållstypen. Vyn Klubbar är därför uppdelad i tre listningssidor där den enda skillnaden är filtreringen av typ-fältet för innehållstypen Klubbar/Organ/Utskott. Varje listningssida filtrerar inläggen efter sin egen typ men använder samma visningsformat.

Colangine-temats *showcase*-element (Figur 18) användes för att ge vyn ett mer dynamiskt utseende. *Showcase* är ett presentationsalternativ byggd med CSS3 som fungerar huvudsakliga elementet på Portfolio-sidorna som Colangines HTML-tema erbjuder. Elementet är uppbyggt av HTML-stuktur som får sitt utseende av Colangines CSS-fil och som inkluderats via *colangine.info*. *Showcase* kan implementeras i två, tre eller fyrakolumners rader. Eftersom det tema som implementeras för SHS.fi har ett konstant sidofält som tar en del av sidutrymmet valdes tre kolumner för *showcase*-vyn.



Figur 18. *Showcase* implementation på Colangines demonstrationsstartsida (Themeforest.net 2014)

För att kunna modifiera en vns utseende kan man inledningsvis ställa in de klasser som placeras i vns rade- och fältelement. *Showcase* kräver att listan omsluts av ett element med klasserna *mbs* och *clearfix* och att den omslutande ``-taggen har klasserna *showcase*, *three-columns*, *sidebar* och *portfolio*. Dessa kan ges direkt genom användargränssnittet i på vns inställningssida under formatinställningarna.

Klubbar: Style options

For All displays

Grouping field Nr.1

- None -

You may optionally specify a field by which to group the records. Leave blank to not group.

Row class

row class

The class to provide on each row. You may use field tokens from as per the "Replacement patterns" used in "Rewrite the output of this field" for all fields.

☒ Add views row classes

Add the default row classes like views-row-1 to the output. You can use this to quickly reduce the amount of markup the view provides by default, at the cost of making it more difficult to apply CSS.

☒ Add striping (odd/even), first/last row classes

Add css classes to the first and last line, as well as odd/even classes for striping.

List type

☒ Unordered list

☐ Ordered list

Wrapper class

mbs clearfix

The class to provide on the wrapper, outside the list.

List class

showcase three-columns sidebar po

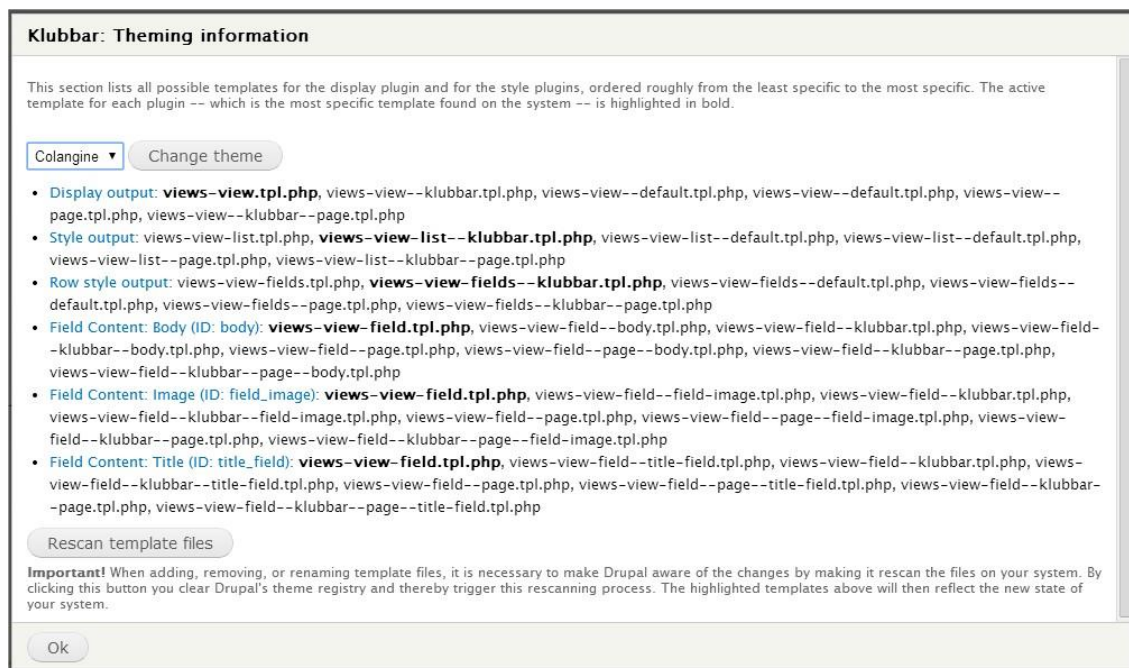
The class to provide on the list element itself.

Apply

Cancel

Figur 19. Klassinställningar för vyformatet

För mer omfattande ändringar bör en eller flera av vyns mallar ersättas i temakatalogen. Ersättningsmöjligheterna för en vy kan härledas på samma sätt som andra mallar. För att förenkla processen finns det även en lista på användbara mallar på vyns inställningssidor under de Avancerade alternativen. I listan visas namngivningsalternativ för mallar, vilka mallar som är i användning och länkar till källkoden för standardmallarna.



Figur 20. Namngivningsalternativ för vyn klubbars mallar

Eftersom *showcase* behöver attributen *data-id* och *data-type* samt en specifik klass i HTML-listans listelement bör den mall som genererar vyns ``-taggar ersättas. Efter inspektion av källkoderna i listan fastslogs det att det var fråga om mallen *views-view-list.tpl.php*. För att ersätta denna mall för samtliga displayer specificerades mallens namn ytterligare med vyns namn och den nya mallen namngavs således *views-view-list--klubbar.tpl.php*. För enkelhetens skull lönar det sig att kopiera den standardkällkod som visas för ursprungsmallen och sedan klistra in och modifiera koden därefter.

Själva innehållet i resultatens ``-taggar är fortfarande det som vyn skriver ut som standard. Fälten i elementet skrevs ut i mallen *views-view-fields.tpl.php* och för att ändra på strukturen ersattes den med mallen *views-view-fields--klubbar.tpl.php*. Strukturen av ett list-element för *showcase* är följande.

```

<li class="color8" data-id="id-1" data-type="digital">
  <a href="länk till innehållet">
    
    <h3>
      Rubrik <i class="icon-angle-right"></i></h3>
    <p>"Brödtext från inlägget"</p>
  </a>
</li>

```

Figur 21. Strukturen av ett list-element för showcase

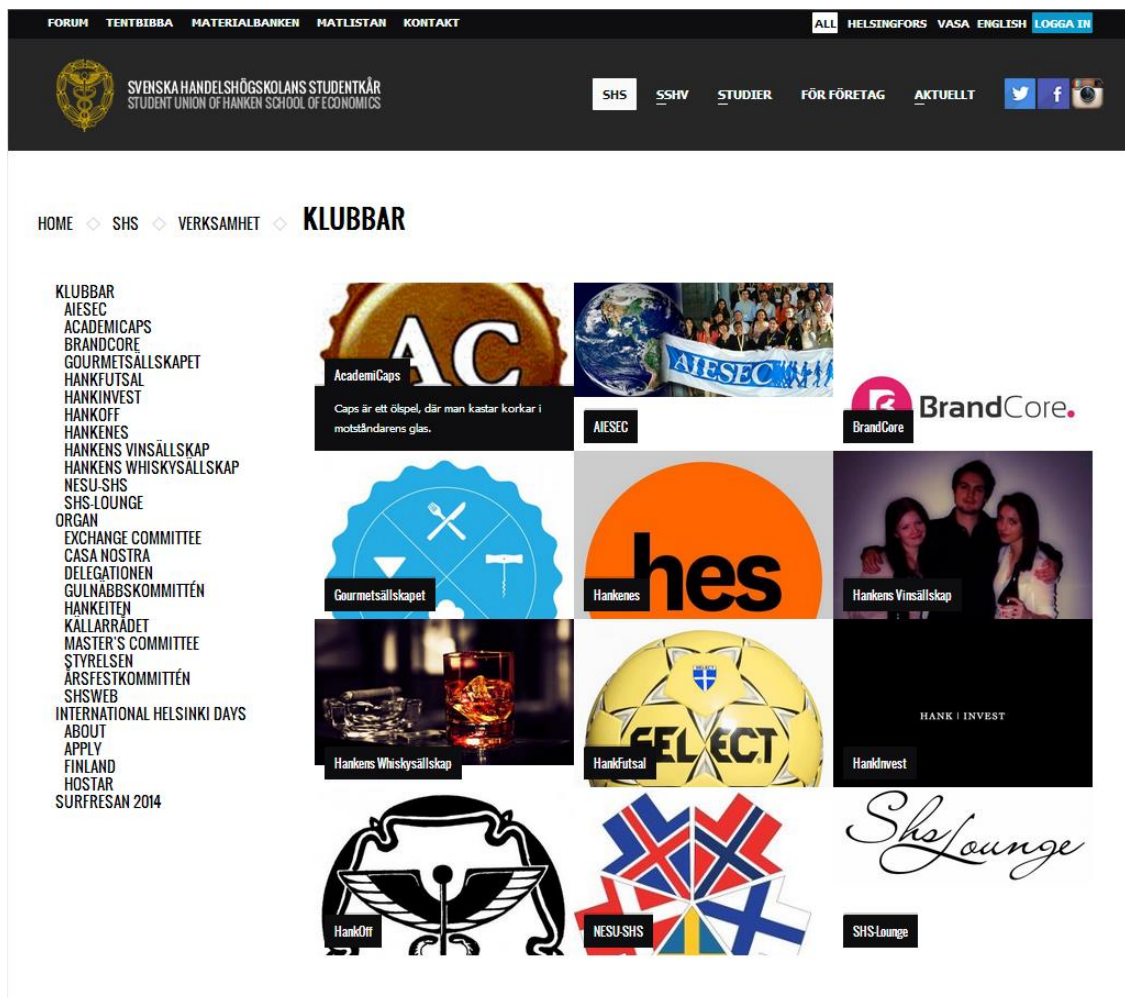
Huvudfunktionen för *views-view-fields.tpl.php* är att iterera igenom de fält som specificerats för vyn. Samtliga fält har sin egen mall och det är möjligt att omstrukturera innehållet i dessa efter behov. Eftersom *showcase* inte klarar av att fler fält läggs till utan att fler strukturförändringar krävs valdes det att hindra *views-view-fields--klubbar.tpl.php* från att iterera genom fälten utan i stället direkt skriva ut den eftertraktade strukturen och ignorera fältens specifika mallar (Figur 22). All information för det ifrågavarande resultatet i listan går att komma åt genom variabeln *row* och med den kan mallens dynamiska innehåll förses.

```

<a href="<?php print $base_path; ?>node/<?php print $row->nid; ?>">
  <?php print theme_image_style(array(
    'style_name' => 'large',
    'path' => $row->field_field_image[0]['raw']['uri'],
    'width' => null,
    'height' => null)
  ); ?>
  <h3><?php print $row->field_title_field[0]['rendered']['#markup'] ?></h3>
  <?php print $row->field_body[0]['rendered']['#markup']; ?>
</a>

```

Figur 22. Mallen *views-view-fields--klubbar.tpl.php*



Figur 23. Listningssida som implementerar showcase-presentation

5.2.3 Blogg

Innehållstypen blogg är skapad för att användas som ett flöde för organisationsrelaterade meddelanden. Fälten för innehållstypen är följande:

- Title (Text)
- Author (Enhetsreferens)
- Blog header image (Bild)
- Body (Lång text och sammandrag)
- Tags (Term referens)


Fältet Author är obligatoriskt och är länkat till innehållstypen Klubbar/Organ/utskott. Blogginlägg är således alltid publicerade av en organisation som har en kontaktsida på hemsidan. Funktionaliteten för enhetsreferenser förses av modulen *Entity reference*.

Tags-fältet är länkat till vokabulären Tags och ger automatiskt förslag på existerande kategorier för innehållet. Om det värde som skrivits in i fältet inte existerar i vokabulären skapas en ny taxonomi med det ifrågavarande värdet.



Innehållssidan (Figur 24) för blogginlägg baserar sig på Colangine-temats *blog-single*-mall. En ny mall *node--blog-entry* skapades och mallen modifierades för Drupal-bruk (Figur 25). På sidan visas blogginläggets bild följt av inläggsinformationen, blogginlägget och taggarna för det ifrågavarande inlägget. Kommenteringen av blogginlägg sker genom en tillämpning av Facebook-kommentering. Sidofältet på bloggsidan visar upp en lista på de senaste blogginläggen.

FORUM
TENTBIBBA
MATERIALBANKEN
MATLISTAN
KONTAKT

ALL
HELSTINGFORS
VASA
ENGLISH
LOGGA IN


SVENSKA HANDELSHÖGSKOLANS STUDENTKÅR
 STUDENT UNION OF HANKEN SCHOOL OF ECONOMICS

SHS
SSHV
STUDIER
FÖR FÖRETAG
AKTUELLT

HOME
◇
AKTUELLT
◇
BLOG
◇
STYRELSEKAFFE 16.4!

SENASTE

MC ORDFÖRANDE SÖKES!
BY STYRELSEN | 10.04.2014

STYRELSEKAFFE 16.4!
BY STYRELSEN | 10.04.2014

STYRELSENS MÖTESREFERAT #16 ÅR 2014
BY STYRELSEN | 10.04.2014

HANKINVEST LONDON EXCURSION
BY HANKINVEST | 09.04.2014

SSHV-S TUTORER 2014
BY SHSWEB | 08.04.2014

STUDENTREPRESENTANT TILL INSTITUTIONSRÅDET FÖR REDOVISNING OCH HANDELSRÄTT (DL 17.4.2014)
BY STYRELSEN | 08.04.2014

MATRÅDET HAR EN LEDIG STUDENTREPRESENTANTSPATS (DL 15.4.2014)
BY STYRELSEN | 08.04.2014

STIPENDIER FÖR UTBYTESSTUDIER
BY STYRELSEN | 07.04.2014

INFORMATION KRING NYA SIDORNA
BY SHSWEB | 07.04.2014

HANKENS VINSÄLLSKAPS ITALIEN-RESA
BY HANKENS VINSÄLLSKAP | 06.04.2014







Svenska Handelshögskolans



STYRELSEKAFFE 16.4!
 Styrelsen

Hej, som fortsättning till "Fråga styrelsen dagen" som ordnades på shsweb i februari så ordnar vi nu följande evenemang i Hankens foajé 16.4. Vi hoppas att alla kommer och talar med oss, och säger rakt ut sina åsikter om styrelsens verksamhet det här året hittills, och vad vi borde göra bättre på hösten. Vi uppskattar både ris och ros, så berätta oss gärna vad vi ska fortsätta på samma sätt med.

Vi kommer att stå i foajén från klockan 11 till klockan 13, och kommer att ha en whiteboard alla kan skriva sina åsikter på. Vi kommer också att ha en låda dit man kan lägga anonyma förslag, så man kan ge feedback också namnlöst.

Ni får också gärna ge oss förslag vad studentkåren borde sträva efter, och hur vi gör SHS till en bättre studentkår för varje medlem.

styrelsen styrelsekafe

Figur 24. Blogginlägg på webbplatsen


```

<div class="post clearfix">
  <?php if(!empty($field_image)): ?>
    <div class="thumbnail">
      <?php print theme_image_style(array(
        'style_name' => 'blog_header_image',
        'path' => $field_image[0]['uri']
      )); ?>
    </div>
  <?php endif; ?>
  <div class="post-title-row clearfix">
    <?php print $date; ?>

    <h3 class="post-title"><?php print $title; ?></h3>

    <div class="meta-more">
      <span><i class="icon-user"></i>
        <?php print l(
          $field_author[0]['entity']->title,
          'node/'. $field_author[0]['entity']->nid
        ); ?>
      </span>
    </div><!-- meta date -->
  </div><!-- title -->

  <div class="post-content">
    <?php print render($content['body']); ?>
    <?php print render($content['field_tags']); ?>

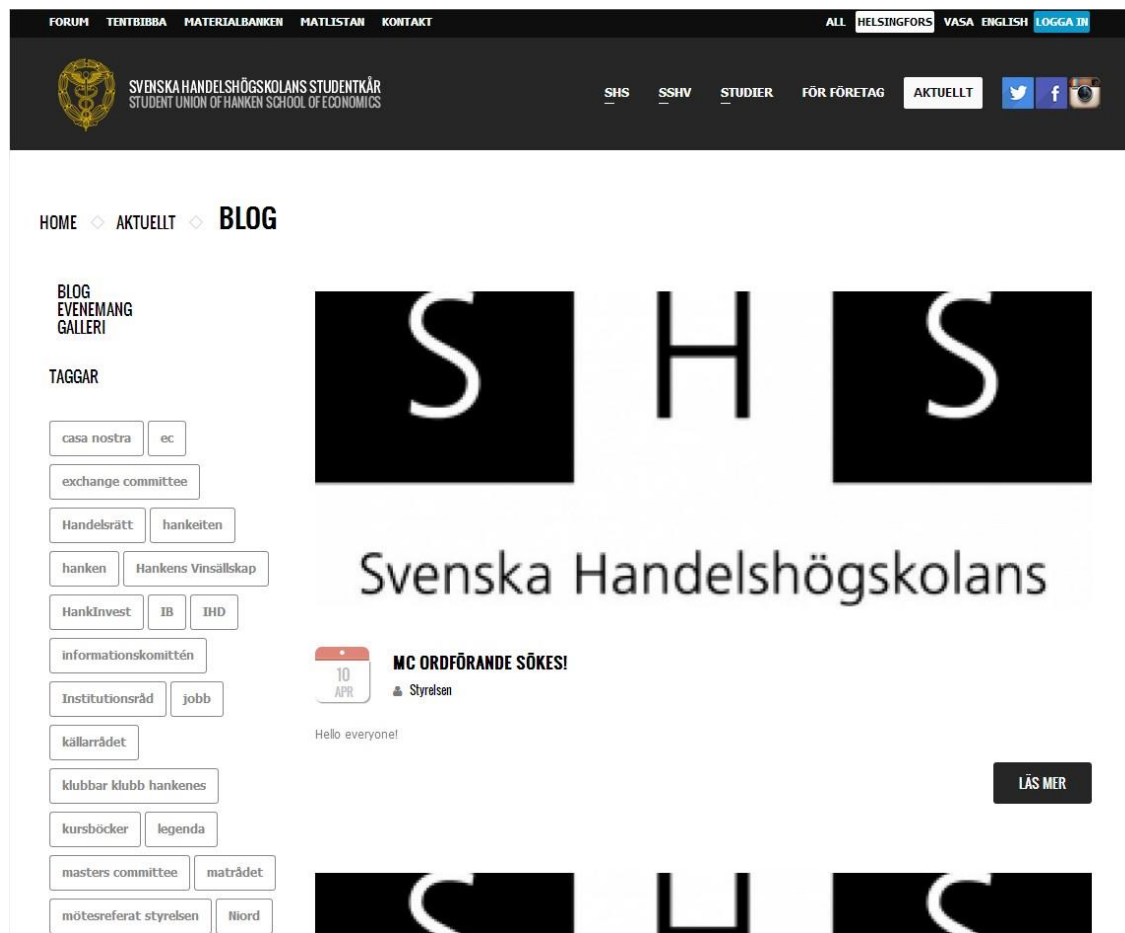
  </div><!-- post content -->

  <?php if(isset($content['facebook_comments'])): ?>
    <?php print render($content['facebook_comments']); ?>
  <?php endif; ?>
</div>

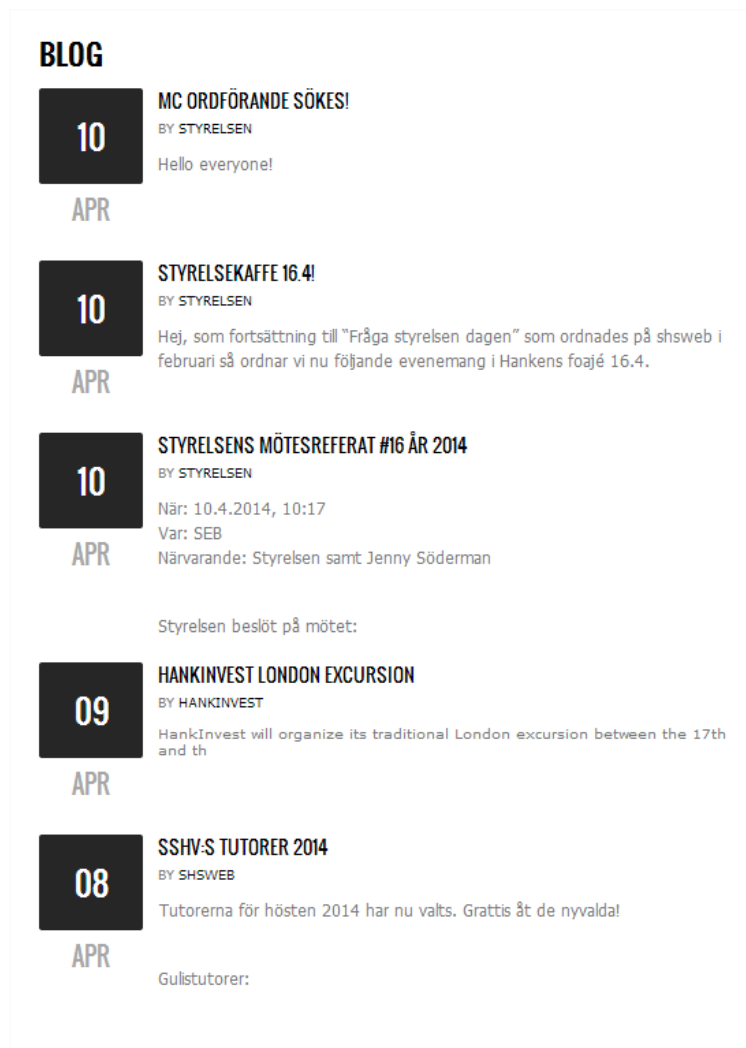
```

Figur 25. Mallen node--blog-entry.tpl.php

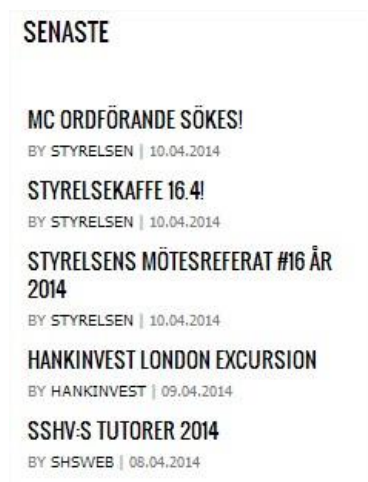
Elementen i huvudvyn för blogginlägg innehåller i huvudsak samma element som den enskilda blogginlägget (Figur 26). För bättre överskådlighet trimmades inläggets text ner till 200 tecken och en ”Läs mer”-knapp lades till. Genom att lägga till taggar som ett filterkriterium i vyinställningarna och definiera kriteriet som exponerat gavs användarna möjlighet att själv filtrera bloggarna. För bättre presentation av de exponerade alternativen användes modulen *Better Exposed Filters*. Övriga listningar av bloggar skapas av samma vy som listningssidan. Framsidans listning (Figur 27) komprimerades genom att skala av taggarna och inläggets bild och i sidofältslistningen (Figur 28) visas endast rubriken, organisationen och publiceringsdatumet.



Figur 26. Huvdlistningen för bloggar



Figur 27. Framsidans blogglistning



Figur 28. Sidofältslistning av bloggar

Inläggsinformationen i Colangine-temats blogginlägg visar upp datumet med ett element som krävde en specifik HTML-struktur. Strukturen varierar beroende på vilken av displayerna som inlägget visas på. För att skriva ut denna struktur skapades en förprocesserare för noder i *template.php*. Förprocesseraren förser mallar med en ny variabel för datumpresentationen då innehållstypen för noden är blogg. Variabelns innehåll byggs upp av en egen funktion, *build_date_markup*, som returnerar HTML-strukturen för datumpresentationen för den vydisplay som definierats (Figur 29).

```
function build_date_markup($date, $display = null)
{
    if (strpos($date, '-') > 0) {
        $date = strtotime($date);
    }

    if ($display == 'node_page' || $display == 'page') {
        // Markup for blog listing and blog-single page.
        $output = '<div class="meta-date">
            <span class="day">' . date('d', $date) . '</span>
            <span class="month">' . date('M', $date) . '</span>
        </div>';
    } else {
        // Markup for blog- and event listings.
        $output = '<div class="grid_2 alpha">
            <span class="date">' . date('d', $date) . '</span>
            <span class="month">' . date('M', $date) . '</span>
        </div>';
    }

    return $output;
}
```

Figur 29. Funktionen *build_date_markup*.

5.2.4 Evenemang


Denna innehållstyp skapades för att göra det möjligt att skapa evenemang för händelsekalendern. Innehållstypen är ämnad att förse en läsare med all viktig information för ett evenemang. För att informationen ska vara enkel att hitta skapades ett flertal specifika fält för att evenemangens information skulle kunna skrivas ut på ett mer uniformt sätt. Fälten för innehållstypen är följande:

- Title (Text)




- Var (Text)
- När (Datum)
- Body (Lång text och sammandrag)
- Pris (Text)
- Biljettförsäljning (Text)
- Tema (Text)
- Dresscode (Text)
- Arrangör (Text)
- e-mail (Text)
- Relaterade sidor (Entitetsreferens)

Innehållssidans visningselement är uppdelade på samma sätt som klubbsidorna. Bilden för evenemanget, brödtexten samt fältet för Facebook-kommentarer visas i sidans innehållselement medan kontakt- och evenemangsinformationen visas i sidofältet genom *node_info*-variabeln (Figur 30). För att fälten ska läggas till i variabeln rendererades samtliga fält i förprocesseringsfunktionen och placerades i variabeln (Figur 31).

FORUM
TENTBIBBA
MATERIALBANKEN
MATLISTAN
KONTAKT
ALL
HELSINGFORS
VASA
ENGLISH
LOGGA IN



SVENSKA HANDELSHÖGSKOLANS STUDENTKÅR
STUDENT UNION OF HANKEN SCHOOL OF ECONOMICS

SHS
SSHV
STUDIER
FÖR FÖRETAG
AKTUELLT

HOME
AKTUELLT
EVENEMANG
HERR- OCH DAMMIDDAG 2014

När: 12.04.2014 - 19:00
Var: Casa
Pris: 20€
Biljettförsäljning:
Tema:
Dresscode:
Arrangör: CN
email: styrelsen@shs.fi



*Datum: 12 april
Plats: Hanken & Casa
Biljettsläpp: 7/4 Kl. 12.00 i fogen
Pris: 20 €
Tid: Coccoare 19.00, sllg 20.00
Tema: Monte Carlo
Max 2 biljetter/Hankenit*

Den eftertraktade Herr- & Dammiddagen är äntligen här!

Evenemanget äger rum lördagen den 12 april kl. 19. Herrarna sllzar på Casa och damerna på Hanken. Den gemensamma efterfesten hålls sedan

Figur 30. Evenemanginlägg på webbplatsen

```

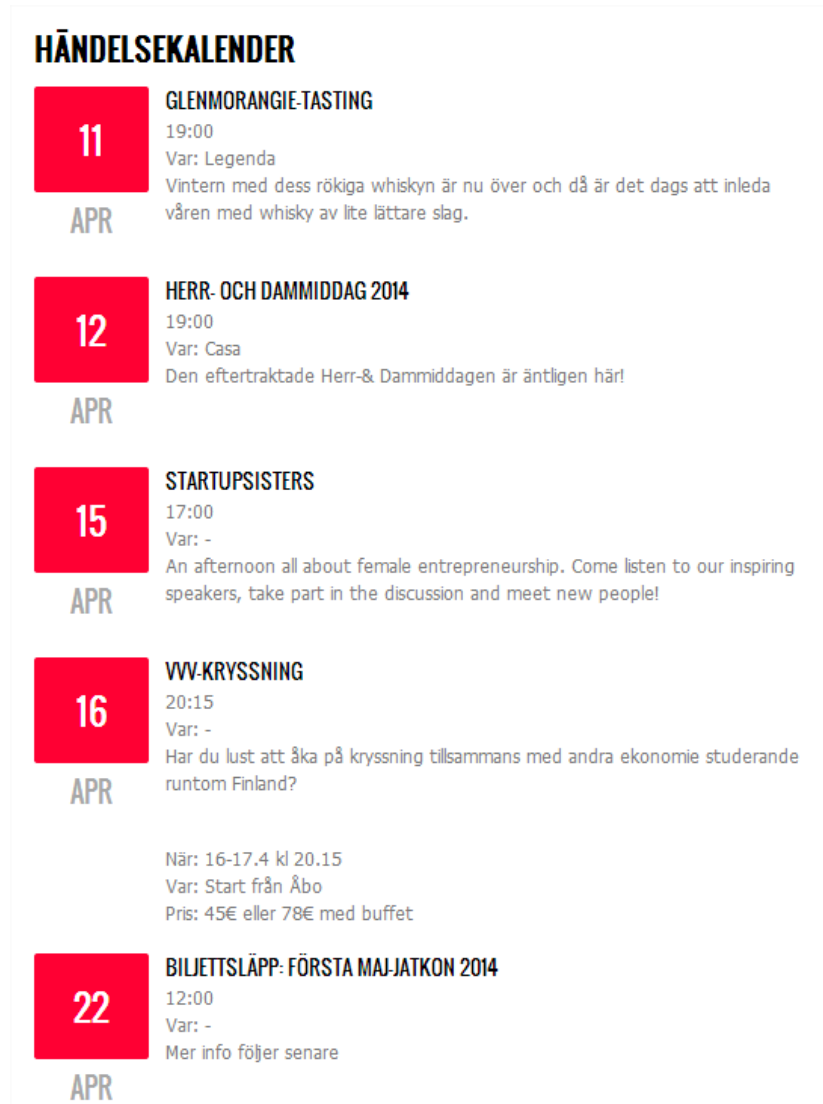
case 'evenemang' :
    $node_info = drupal_render(field_view_field('node',$node, 'field_nar', 'default'));
    $node_info.= drupal_render(field_view_field('node',$node, 'field_var', 'default'));
    $node_info.= drupal_render(field_view_field('node',$node, 'field_pris', 'default'));
    $node_info.= drupal_render(field_view_field('node',$node, 'field_biljettf_rs_ljning', 'default'));
    $node_info.= drupal_render(field_view_field('node',$node, 'field_tema', 'default'));
    $node_info.= drupal_render(field_view_field('node',$node, 'field_dresscode', 'default'));
    $node_info.= drupal_render(field_view_field('node',$node, 'field_arrangor', 'default'));
    $node_info.= drupal_render(field_view_field('node',$node, 'field_email', 'default'));
    $node_info.= drupal_render(field_view_field('node',$node, 'field_campus', 'default'));
    break;

```

Figur 31. Inplacering av fält i node_info-variabeln

Vyn ”Händelsekalender” är i första hand byggt som ett block för framsidan som visar fem kommande evenemang för en månad framåt. Presentationen av händelsekalendern är den samma som den för de komprimerade blogginläggen. Mallen views-view-fields.tpl.php ersattes av views-view-field--event_calendar. Den nya mallen var ursprungligen den samma

som för den komprimerade blogglistan som sedan ändrades till att visa upp den information som evenemangen hade att erbjuda. Huvudlistningen av evenemang skiljer sig inte från framsidans block utseendemässigt utan har endast getts en vidare bredd och en bläddrare för att se alla kommande evenemang (Figur 32).



Figur 32. Framsidans händelsekalender

5.2.5 Galleri

Innehållstypen Galleri skapades för att tillåta uppladdning av ett flertal bilder för ett bildalbum. Fälten för innehållstypen är:

- Title (Text)

- Body (Lång text och sammandrag)
- Bilder (Bild)
- Datum (Datum)
- Fri Text (Lång text)

Fältet Bilder använder sig av modulerna *Multiupload Filefield Widget* och *Multiupload Filefield Widget* för att möjliggöra samtidig uppladdning av ett större antal bilder. För att fältet skulle fungera felfritt krävdes även att en del ändringar i php-inställningarna för maximal filuppladdningsstorlek och antal på produktionsservern. Eftersom *Multiupload Filefield Widget* laddar upp samtliga filer som ett gemensamt paket blir filstorleken summan av alla de filer som önskas ladda upp då innehållet sparas.

Inläggssidan har ingen huvudbild så brödtexten kommer överst i sidans innehållspresentation. Under brödtexten skrivs albumets samtliga bilder ut i rader. Längst nere på sidan finns möjligheten till Facebook kommentarer.

För förstöringsfunktionalitet samt bildspel använder galleriet jQuery-biblioteket prettyPhoto som var inkluderat i Colangine-temat. För denna funktionalitet krävdes omstrukturering av bildfältet och mallen *node.tpl.php* måste ersättas med *node--galleri.tpl.php*. Mallen itererar genom samtliga bilder i galleriet och placerar dem i en `<a>`-tagg med attributet *data-gal* och klassen *gallery-item* (Figur 33).

```
<div class="gallery">

<?php foreach($node->field_gallery_images[LANGUAGE_NONE] as $item): ?>
  <a href="<?php print file_create_url($item['uri']);?>" data-gal="photo[gallery]" class="gallery-item">
    <?php print theme_image_style(array(
      'style_name' => 'thumbnail',
      'path' => $item['uri'],
      'width' => null,
      'height' => null)); ?>
  </a>
<?php endforeach; ?>

</div>
```

Figur 33. Uppbyggnad av HTML-stukturen för implementering av prettyPhoto

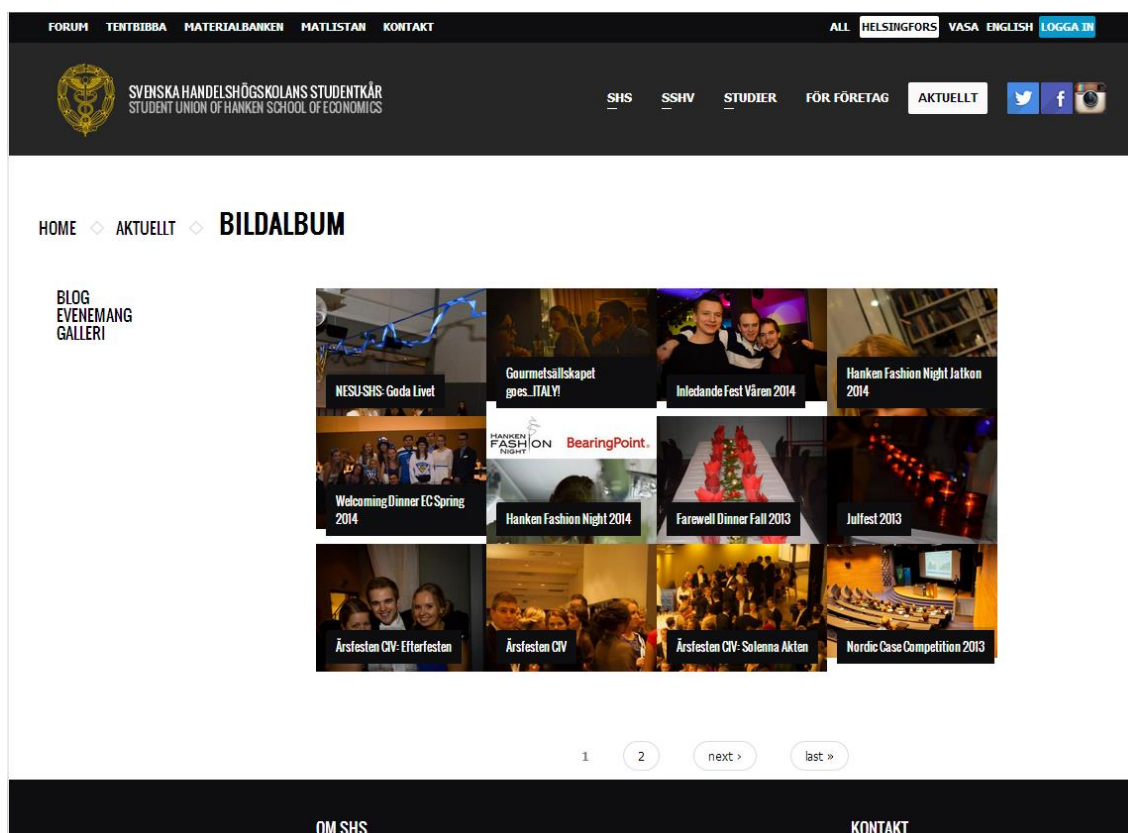
Listningen av galleriinläggen använder sig av samma *showcase*-presentation som Klubb/Organ/Utskott. För implementationen kopierades de mallar som använts för Klubb/Organ/Utskott och namngavs om för att implementeras i galleri vyn. I filen views-

view-fields--gallery.tpl.php var variabeln för visningsbildens variabel tvungen att ändras eftersom galleri inte använder samma fält för bilder som Klubb/Organ/utskott (Figur 34).

```
<a href="<?php print $base_path; ?>node/<?php print $row->nid; ?>">
<?php print theme_image_style(array(
  'style_name' => 'large',
  'path' => $row->field_field_gallery_images[0]['raw']['uri'],
  'width' => null,
  'height' => null)
); ?>
<h3><?php print $row->field_title_field[0]['rendered']['#markup'] ?></h3>
<?php print $row->field_body[0]['rendered']['#markup']; ?>
</a>
```

Figur 34. Återanvändning av views-view-fields--klubbar.tpl.php som views-view-fields--gallery.tpl.php

Vyn plockar ut den bild som hittas först i listan av bilder som laddats upp i Bilder-fältet. I *showcase*-elementets stigande textfält visas brödtexten för inlägget. Vyn har två displayer där en är ett block som visar upp sex element och en annan är listningssidan för galleriinlägg (Figur 35) där tolv element visas och en bläddrare ger möjlighet att komma åt äldre gallerier.



Figur 35. Huvudlistning för galleriinlägg

5.2.6 Forum

Forumet på shs.fi är en direkt implementation av Drupals standardmodul *Forum*. Modulen är uppbyggd förser Drupal-installationen med elementen *Forum container*, *Forum* och *Forum topic*.

Komponenten *Forum topic* är innehållstypen som alla användare kan skapa och kommentera. Användaren ger inlägget en rubrik och brödtext och väljer ett *Forum* att publicera inlägget i. Övriga användare har möjlighet att kommentera inlägget. För att ordna *Forum* enligt ämne är det även möjligt att definiera *Forum containers* som dessa kan organiseras i. Till skillnad från *Forum* kan *Forum topics* inte skapas i *Forum containers*, däremot kan *containers* placeras inuti *Forum*. (Drupal.org 2014m)

Huvudsidan för forum är roten för forumstrukturen. På sidan visas alla *Forum* och foruminlägg på hela forumets rotnivå i en tabell. Colangine-temats *shortcodes* innehåller ett alternativ för utseendet på tabeller vilket valdes att implementeras på forumsidorna. För detta krävdes att klassen *table* finns definierad i tabellens `<table>`-tag. Denna möjlighet att påverka tabellelementets klasser finns inte i Drupals användargränssnitt så *forum-list.tpl.php* och *forum-topic-list.tpl.php* ersattes och klasserna placerades in efter behov (Figur 36).

```
<table id="forum-topic-<?php print $topic_id; ?>" class="table">
```

Figur 36. Definiering av ny klass i den ersatta mallen *forum-topic-list.tpl.php*.

För att reducera spam i forumen användes den kontribuerade modulen *CAPTCHA*. Modulens ändamål var att kontrollera att användaren var mänsklig då denne försökte spara Foruminlägg och kommentarer. Lösningen visade sig efter publiceringen av webbplatsen inte vara tillräckligt effektiv för att motverka robotar så modulen *reCAPTCHA* användes för att utöka funktionaliteten.

Kommentarerna för foruminläggen baserades i viss mån efter Colangines mallar men eftersom SHS.fi inte stöder användarbilder skulle en direkt implementation blivit överdådig. För att modifiera kommentarernas innehåll förprocesserades kommentarerna via funktionerna *colangine_preprocess_comment* och *colaninge_preprocess_comment_wrapper*. I funktionerna lades önskade klasser till i variabelistan och i *colangine_preprocess_comment*

skapades variabeln *comment_created* för att formatera utskriften för kommentarens publiceringsdatum. För att skriva ut publiceringsdatumet och implementera den struktur som önskades för kommentarerna ersattes mallen *comment.tpl.php* modifierades enligt mallen.

För att enklare moderera forumet skapades flaggan *spam* för kommentarer med den kontribuerade modulen *Flags*. Flaggan blev global och vid flaggningen av en kommentar opublicerades kommentaren. Opubliceringen sköttes av den kontribuerade modulen *Rules* som ger möjlighet att låta Drupal lyssna efter specifika händelser och vid avfyrad händelse utföra en handling. För spam flaggan valdes att vid flaggning opublicera flaggat innehåll. Flagningsalternativet skrevs ut i *comment.tpl.php*.

5.2.7 Tentbibliotek

Tentbiblioteket är innehållstypen som förser materialbanken och tentbibban med material. Ett inlägg består i huvudsak av filer. Fälten för innehållstypen är följande:

- Kursnamn (Text)
- Body (Lång text och sammandrag)
- Material (Fil)
- Kurs (Term referens)
- Materialtyp (Lista (text))

Eftersom ett inlägg motsvarar en kurs eller ett en typ av material kan ett flertal filer tillskrivas ett inlägg. Eftersom att det redan fanns samlat ett stort antal material som skulle flyttas över till tentbibban och materialbanken krävdes det att ett flertal filer skulle kunna laddas upp på en gång. För denna funktionalitet återanvändes modulen *Multiupload Filefield Widget* som också användes för innehållstypen *Galleri*. För att skilja åt inlägg för tentbibban och materialbanken skapades fältet *Materialtyp* där ändamålet kan definieras.

För kategorisering av innehållet skapades en ny taxonomivokabulär kallad *Kurser*. Genom att i kursfältet fylla i namnet på den enhet innehållet ska kategoriseras under sparas värdet som en term i vokabulären. Fältet är självkompleterande så de termer som redan finns i vokabulären kommer att föreslås för fältet.

Visningen av ett enskilt inlägg består av body-fältets brödtext och listan av de filer som inlägget försätts med. För att komma åt ett inlägg går användaren igenom tre steg. Först val av materialbank eller tentbibba, sedan val av kurs- eller materialkategori och slutligen val av innehållet. Tentbibban och materialbanken är element i sekundärmenyn i sidhuvudet. Dessa länkar till var sin lista av termer som i sin tur länkar till listor av inläggen som har den valda termen tillskriven.

Listningsvyerna för tentbibban respektive materialbanken visar endast de termer som har använts som kategorier för den materialtyp som listningssidan är ämnad för. Denna filtrering åstadkoms genom att i vyninställningarna skapa en relation mellan den term som hanteras med inlägg som utnyttjar fältet *kurs*. Detta hämtar in samtliga inlägg av innehållstypen Tentbibliotek. För att endast lista de kategorier som använts av ifrågavarande materialtyp filtreras resultatet genom att endast tillåta material med typen i fråga. Detta resulterade i upprepningen av ett kategorielement för varje inlägg det tillskrivits och resultatet blev ett mångduplikat. För att filtrera bort de extra resultaten valdes det under *Query settings* i vyns avancerade inställningar att endast distinkta resultat skulle listas.

Genom att länka resultaten i vyn till en ny listningssida med termnamnet som en del av sökvägen kunde termnamnet passeras som ett representativt värde för vyn. Detta betydde att resultatet för vyn kunde filtreras med ett kontextuellt filter så att endast material med den kategori som länkat till sidan visades på listningssida. Även denna vy delades upp i två listningssidor där den ena endast visade upp materialbankens resultat och den andra tentbibbans. Listans innehåll bestod således av länkar till samtliga inlägg som hittats under den valda kategorin.

Eftersom resultaten i bägge vyerna endast var en lista på titlar behövdes ingen extra kod i temakatalogen för att tillämpa presentationen av innehållet. Den eftertraktade presentationen kan åstadkommas direkt via användargränssnittet.

Materialbanken och tentbiblioteket är endast nåbara för inloggade användare. Åtkomsten nekades med hjälp av den kontribuerade modulen *Nodeaccess* som ger möjlighet att hantera

åtkomst för olika användarroller och innehållstyper. Vyerna ställdes in så att de endast visas för inloggade användare.

5.2.8 Banners

För att kunna länka till sidor som inte fanns representerade som noder på SHS.fi krävdes en innehållstyp vars huvuduppgift var att producera en bildlänk. Dessa bildlänkars användningsområde var de sidlyft och sponsorlänkar som visades på framsidan och Aktuellt sidan. Eftersom de vyer som använde sig av bildlänkarna i regel var promotionsartade blev innehållstypen en samlingstyp för promotionsmaterial som kan användas i samtliga promotionsvyer. Fälten för innehållstypen blev på grund av användningsområdet relativt få:

- Title (Text)
- Bild (Bild)
- Link (Länk)
- Beskrivning (Text)

Eftersom användningsområdet för innehållstypen exklusivt var vyer gjordes inga ändringar för innehållstypens enskilda inläggs visningssida.

5.3 Startsidan

Den *page*-mall som vid grundimplementationen skapades för framsidan var en direkt kopia av Colangine-temats *index.html*-fil även om den var implementerad för Drupal-installationen. Detta betydde att alla element på framsidan som visade upp innehåll var statiska och kunde inte påverkas via användargränssnittet. Layouten för framsidan som kommits överens om motsvarade inte heller ursprungsmallen så de gamla elementen byttes i tur ordning ut mot vy-block som visade upp innehåll efter specifikation i användargränssnittet.

Det bildspel som Colangine-temat använt på framsidan var av en typ som inte stöddes av Drupals-användargränssnitt, inga kontribuerade moduler hittades som skulle ha implementerat bildspelet i fråga. Däremot hittades modulen *FlexSlider* som erbjuder ett visningsformat för

View-modulen som gör det enkelt att manipulera ett bildspel via Drupals användargränssnitt. En ny vy konfigurerades för att visa upp ett inläggs bild som länkade till det egentliga inlägget. Inläggen som skulle visas i bildspelet valdes genom ett nytt fält som skulle skapas för varje innehållstyp.

Efteråt specificerades att bildspelets element även skulle kunna länka till sidor utanför Drupal-installationen. Innehållet för bildspelet reglerades då endast till innehållstypen Banners där absoluta länkar kunde förses åt inlägget. För att välja vilka inlägg skulle visas i bildspelet valdes i stället för ett nytt fält en global flaggning av innehållet med hjälp av den kontribuerade modulen *Flags*. Orsaken till bytet av valmetod var att flaggor enklare än fält kunde definieras för ett flertal innehållstyper. Vyn ställdes slutligen in att endast inkludera de bannerinlägg som flaggats med flaggan *slider*. Slidern placerades som ett block i en ny region *hero* som först definierades i *colangine.info*-filen och sedan rendererades i *page--front.tpl.php*-mallen.

Vyn *Banner* under bildspelet placerades också i en nydefinierad region kallad *frontpage_banner*. Vyn var följande implementation av innehållstypen *Banners*. En ny global flagga, *banner*, skapades och associerades med innehållstypen för att kunna specificera vilka inlägg som skulle inkluderas i vyn. Banner vyn visar upp tre bannerinlägg i form av inläggens bilder länkade till respektive material. Fältypen *Link* som används av bannerinläggen kan inte associeras med bildmaterialet ur samma inlägg i vynes användargränssnitt. För att skapa de länkade bilderna ersattes vynes fält-mall med mallen *views-view-fields--banner.tpl.php*. I den nya mallen placerade ett enskilt resultat i ett *<div>*-element med en klass, *third*, som ställdes in att uppta en tredjedel av det omslutande elementets bredd. För själva länken användes Drupals länkfunktion *l* (Drupal.org 2014n). För funktionens textparameter användes funktionen *theme_image_style* för att renderera vynes bildelement som den visuella representationen för länken, för sökvägsparametern användes länken som vyn plockat från bannerinlägget (Figur 37).

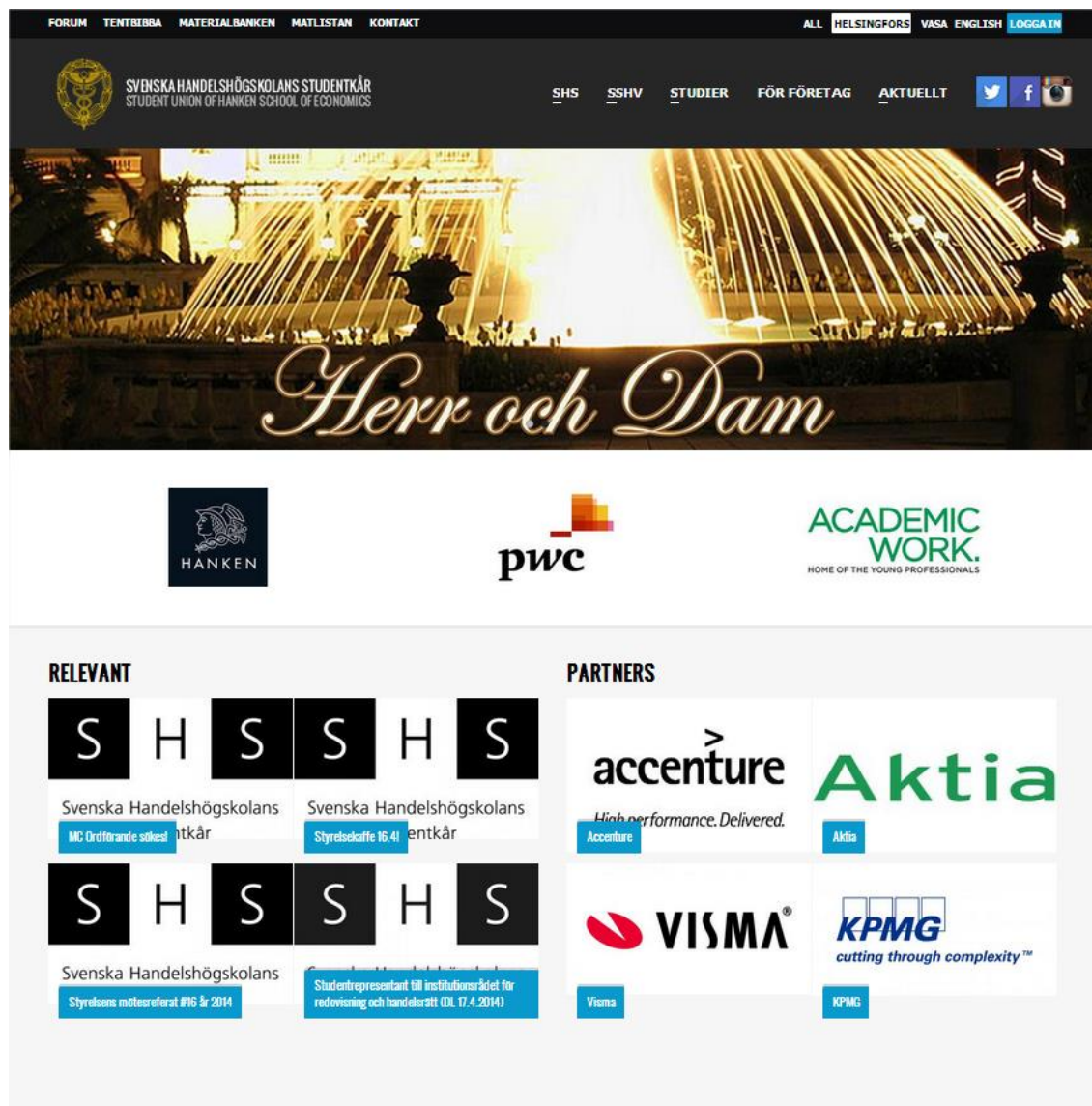
```

<div class="third mbf">
  <?php print l(
    theme_image_style(array(
      'style_name' => 'large',
      'path' => $row->field_field_bild_1[0]['raw']['uri'],
      'height' => null,
      'width' => null
    )),
    $row->field_field_link_1[0]['raw']['url'],
    array(
      'html' => TRUE,
      'absolute' => TRUE,
      'attributes' => array('target' => '_blank'),
    ); ?>
</div>

```

Figur 37. Innehållet för mallen *views-view-fieds--banner.tpl.php*

Under startsidans banner element skapades två bredvidliggande regioner. För regionerna skapades en vy med en display för administratorvalda relevanta inlägg och en för partners. Displayen för relevanta inlägg är en promotionsvy för Drupal-installationens egna inlägg. Denna display listade de senaste inläggen som en administrator markerat med en ny flagga, *relevant*. Bägge displayer använder sig av Colangines *showcase*-element och resultatet skrivs ut som en samling bildlänkar med titel och beskrivningstext. Displayen för partners skulle länka till den i fråga varande partners egna hemsidor så länken behövde således vara absolut. Eftersom kriteriet för innehållet i displayen var en bild och en absolut länk valdes det att igen använda sig av innehållstypen *Banners* genom att skapa ett tredje flaggalternativ för innehållstypen, *sponsorer*. *Showcase*-presentationen implementerades i huvudsak på samma sätt som i fallet för vyn klubbar med skillnaden att de variabler som vyn passerade till mallarna hade varierande namn beroende på display (Figur 38).



Figur 38. Startsidan med bildspel, banner, relevanta inlägg och partners

Det sista elementet på startsidan är ett samlingselement för det senaste materialet på Drupal-sidan (Figur 39). De typer av inlägg som listas i elementet är bloggar, evenemang och gallerier. Innehållet i elementet skrivs ut i tre vyer som bäddats in i mallen direkt med funktionen `views_embed_view`. Vyerna som använts är de block som tidigare beskrivits i samband med innehållstypernas struktur.

BLOG

10

MC ORDFÖRÄNDE SÖKES!
BY STYRELSEN
Hello everyone!

APR

10

STYRELSEKAFFE 16.4!
BY STYRELSEN
Hej, som fortsättning till "Fråga styrelsen dagen" som ordnades på shsweb i februari så ordnar vi nu följande evenemang i Hankens foajé 16.4.

APR

10

STYRELSENS MÖTESREFERAT #16 ÅR 2014
BY STYRELSEN
När: 10.4.2014, 10:17
Var: SEB
Närvarande: Styrelsen samt Jenny Söderman

APR

09

HANKINVEST LONDON EXCURSION
BY HANKINVEST
HankInvest will organize its traditional London excursion between the 17th and th

APR

08

SSHV-STUTORER 2014
BY SHSWEB
Tutorerna för hösten 2014 har nu valts. Grattis åt de nyvalda!

APR

11

GLENMORANGIE-TASTING
19:00
Var: Legenda
Vintern med dess rökiga whiskyn är nu över och då är det dags att inleda våren med whisky av lite lättare slag.

APR

12

HERR- OCH DAMMIDDAG 2014
19:00
Var: Casa
Den eftertraktade Herr-& Dammiddagen är äntligen här!

APR

15

STARTUPSISTERS
17:00
Var: -
An afternoon all about female entrepreneurship. Come listen to our inspiring speakers, take part in the discussion and meet new people!

APR

16

VVV-KRYSSNING
20:15
Var: -
Har du lust att åka på kryssning tillsammans med andra ekonomie studerande runtom Finland?

APR

22

BIJUETTSLÄPP- FÖRSTA MAJ-JATKON 2014
12:00
Var: -
Mer info följer senare

APR

GALLERY

MESU-SHS- Goda Livet

Gourmetsällskapet goes...ITALY

Inledande Fest Våren 2014

Hanken Fashion Night Jatkoon 2014

Welcoming Dinner EC Spring 2014

Hanken Fashion Night 2014

Figur 39. Startsidans nedre fält med vyer för bloggar, händelsekalender och gallerier

5.4 Aktuellt-sidans struktur

För en mer specifierad listningssida skapades sidan *Aktuellt* (Figur 40). Sidan skulle förutom nya inlägg lista upp bloggar med specifika ämnen.

För att skapa en nod för sidan i fråga skapades ett nytt inlägg av typen *Basic page*. Eftersom det innehåll som skulle visas på sidan uteslutande skulle bestå av vyer spelade det egentliga innehållet för det nyskapade inlägget ingen roll i sidans utformning. För att ge sidan en ny struktur skapades en ny *page*-mall för att ersätta standardmalle för noden i fråga. För att ersätta page-mallen för noden specificerades en ny mall med nodens identifikationsnummer,

56

den ersättande mallens filnamn blev således *page--node--117.tpl.php*. Eftersom varken sidofältet eller sidinnehållet behövdes på sidan skalades först allting mellan länkstigen och sidfoten bort från mallen. Den avskalade strukturens innehåll ersattes med tre nya regioner. I den översta regionen, *list_page_feature*, användes samma promotionsvy som startsidan använt för att visa relevanta inlägg men med en fullbredds layout. De två andra regionerna, *list_page_left* och *list_page_right*, i element som tog upp var sin hälft av det resterande sidutrymmet.

Vyerna som placerades i de två lägre regionerna var de samma som bäddats in på startsidan. Dessutom skapades två nya displayer för vyn *Blog* som listade bloggar enligt var sin specifika tagg som filter. Regionerna skapades dock för att nya block enkelt skulle kunna läggas till på sidan.

HOME
AKTUELLT

S

H

S

S

H

S

S

H

S

S

H

S

Svenska Handelshögskolans
MC Ordförande sökes! entkär

Svenska Handelshögskolans
Styrelsekafe 16.4! entkär

Svenska Handelshögskolans
Styrelsens mötesreferat #16 År 2014

Studentrepresentant till institutionsrådet för redovisning och handelsrätt (DL 17.4.2014)

Blog

10

MC ORDFÖRANDE SÖKES!

BY STYRELSEN

Hello everyone!

APR

10

STYRELSEKAFFE 16.4!

BY STYRELSEN

Hej, som fortsättning till "Fråga styrelsen dagen" som ordnades på shsweb i februari så ordnar vi nu följande evenemang i Hankens foajé 16.4.

APR

10

STYRELSENS MÖTESREFERAT #16 ÅR 2014

BY STYRELSEN

När: 10.4.2014, 10:17

Var: SEB

Närvarande: Styrelsen samt Jenny Söderman

APR

09

Styrelsen beslöt på mötet:
HANKINVEST LONDON EXCURSION

BY HANKINVEST

HankInvest will organize its traditional London excursion between the 17th and th

APR

08

SSHV-S TUTORER 2014

BY SHSVES

Tutorerna för hösten 2014 har nu valts. Grattis åt de nyvalda!

APR

08

Gullistutorer:

Händelsekalender

11

GLENMORANGIE-TASTING

19:00

Var: Legenda

Vintern med dess rökiga whiskyn är nu över och då är det dags att inleda våren med whisky av lite lättare slag.

APR

12

HERR- OCH DAMMIDDAG 2014

19:00

Var: Casa

Den eftertraktade Herr-& Dammiddagen är äntligen här!

APR

15

STARTUPSISTERS

17:00

Var: -

An afternoon all about female entrepreneurship. Come listen to our inspiring speakers, take part in the discussion and meet new people!

APR

16

VVV-KRYSSNING

20:15

Var: -

Har du lust att åka på kryssning tillsammans med andra ekonomie studerande runtom Finland?

APR

16

När: 16-17.4 kl 20.15

Var: Start från Åbo

Pris: 45€ eller 78€ med buffet

22

BIJETTSLÄPP- FÖRSTA MAJATKON 2014

12:00

Var: -

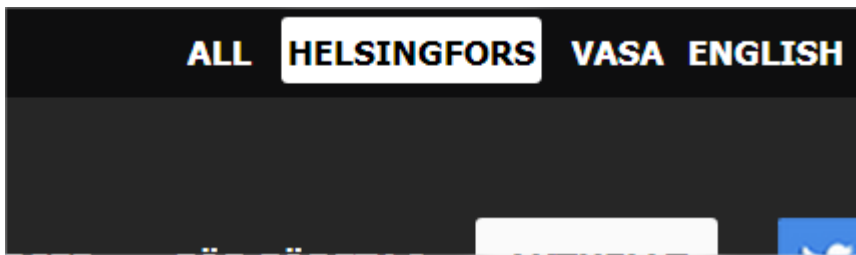
Mer info följer senare

APR

Figur 40. Aktuellt sidan på webbplatsen

5.5 Lokalisation

Eftersom SHS.fi är ämnad för alla studerande vid Svenska handelshögskolan krävde sidan en del filtreringsfunktionalitet. Skolans huvudspråk är svenska och engelska vilket kräver möjlighet till översättning av innehåll (Figur 41). Skolan har även campus både i Helsingfors och Vasa så filtrering av vyerna enligt önskat campus var även specificerat.



Figur 41. Alternativen för lokaliserat innehåll

5.5.1 Språk

För möjligheten till språkalternativ till sidan och dess innehåll användes ett flertal kontribuerade moduler. För basfunktionaliteten för flerspråkigheten används modulen *Internationalization* (i18n). Modulen är ett paket som innehåller ett flertal valfria moduler för språkliga ändamål. Följande av dessa moduler har inkluderats i installationen:

- Block Languages (Språkval för block)
- Field Translation (Översättning av ett fälts egenskaper)
- Internationalization (Utökat stöd för flerspråkig funktionalitet)
- Menu translation (Översättning av menyelement)
- String translation (Översättning av strängar skapade av användare)
- Translation sets (Förser gränssnitt för utomstående moduler)

För att definiera språkvalet åt Drupal-installationen valdes alternativet sökvägsprefix. Detta alternativ gör att Drupal kollar den aktiva URL-adressen efter vilket prefix är i användning och baserar språket efter resultatet. Prefixet för det svenskspråkiga innehållet har lämnats blankt för att hålla svenska som standardspråk medan det engelska prefixet är "en". För att se den engelskspråkiga versionen av framsidan blir URL-adressen www.shs.fi/en och alla övriga sökvägar för engelskt innehåll skrivs ut efter prefixet. Språkvalet erbjuds i form av ett block som tillförts av *i18n*-modulen och plaserats i sidhuvudet.

Då *i18n*-modulens lösning för översättning av inlägg grundar sig på att dela upp innehållet i en nod per språk blir antalet noder snabbt högt på webbplatsen och överskådligheten blir sämre för administratorer. Den kontribuerade modulen *Entity translation* erbjuder en mer attraktiv lösning i och med att innehållstypernas fält kan definieras som översättbara. Denna lösningens slutresultat är en nod per inlägg men det valda språket bestämmer vilken

översättning som skrivs ut i fälten. Titelfältet för en nod är inte av samma typ som övriga fält i Drupal. För att få nodens titel översatt krävdes att den kontribuerade modulen *Title* implementerades. Modulen konverterar valda innehållstypers titlar till ett fält som *Entity translate*-modulen klarar av att översätta. (Drupal.org 2014o)

Översättning av menyelement med *i18n*-modulen har samma svaghet som vid översättningen av inlägg. Ett nytt element måste skapas för samtliga länkar där språket måste väljas. Endast språkneutrala element eller element på det definierade språket visas i menyerna. Modulen *Entity translation menu* är en valfri del av Entity translation modulpaketet som tillför funktionaliteten för att översätta enskilda element. Eftersom Entity translation modulen översätter noder kommer både den svenskspråkiga och engelskspråkiga versionen av ett inlägg dela samma interna sökväg. Eftersom detta är fallet behöver *Entity translation menu*-modulen endast tillföra möjligheten att översätta menyelementets namn efter det aktiva språket. Eftersom en del av innehållet på SHS.fi av praktiska skäl inte gick att översättas med den funktionalitet som *Entity translation* erbjöd behölls *i18n*-modulens översättningsfunktionalitet i de enstaka fallen.

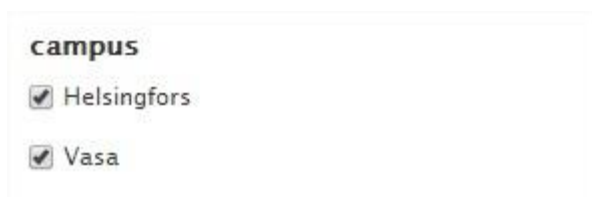
De strängar som skrivs ut i temats mallar som del av koden kräver att översättningsfunktionen *t* används för att översättning av strängen ska vara möjlig. Strängen i temats kod skrivs på engelska och kan översättas liksom alla andra strängar via Drupal's användargränssnitt. Exempelvis kan strängen "Commented" göras översättbar genom att implementera den med *t*-funktionen på följande sätt: *t('Commented')*. Alla strängar som skrivs ut direkt i kod bör använda sig av översättningsfunktionen för att underlätta översättningen. (Drupal.org 2014p)

5.5.2 Campus

Eftersom Svenska handelshögskolan har campus i Helsingfors och Vasa fungerar SHS.fi inte bara som hemsida för Svenska Handelshögskolans Studentkår utan även för Studentföreningen vid Svenska Handelshögskolan i Vasa (SSHV). Eftersom båda

föreningarna lägger upp aktuellt innehåll på SHS.fi specificerades möjligheten att filtrera innehåll på basis av campus.

Eftersom de enskilda sidorna enkelt kan skiljas åt på basis av campus genom att placera dessa under SHS respektive SSHV i menyhierarkin påverkas endast listningarna av den sidomfattande filtreringen. Detta betyder att även om inte sidorna i sig kräver en campusvariabel krävs denna för fallet av att ett inlägg vill lyftas upp på framsidan. Samtliga innehållstyper har därför getts ett *Campus*-fält i form av valet mellan Helsingfors, Vasa eller bägge campus (Figur 42).

The image shows a web form with the title "campus" in bold. Below the title, there are two radio button options. The first option is "Helsingfors" and the second is "Vasa". Both radio buttons are checked, indicating that both campuses are selected. The form is styled with a light gray border and a white background.

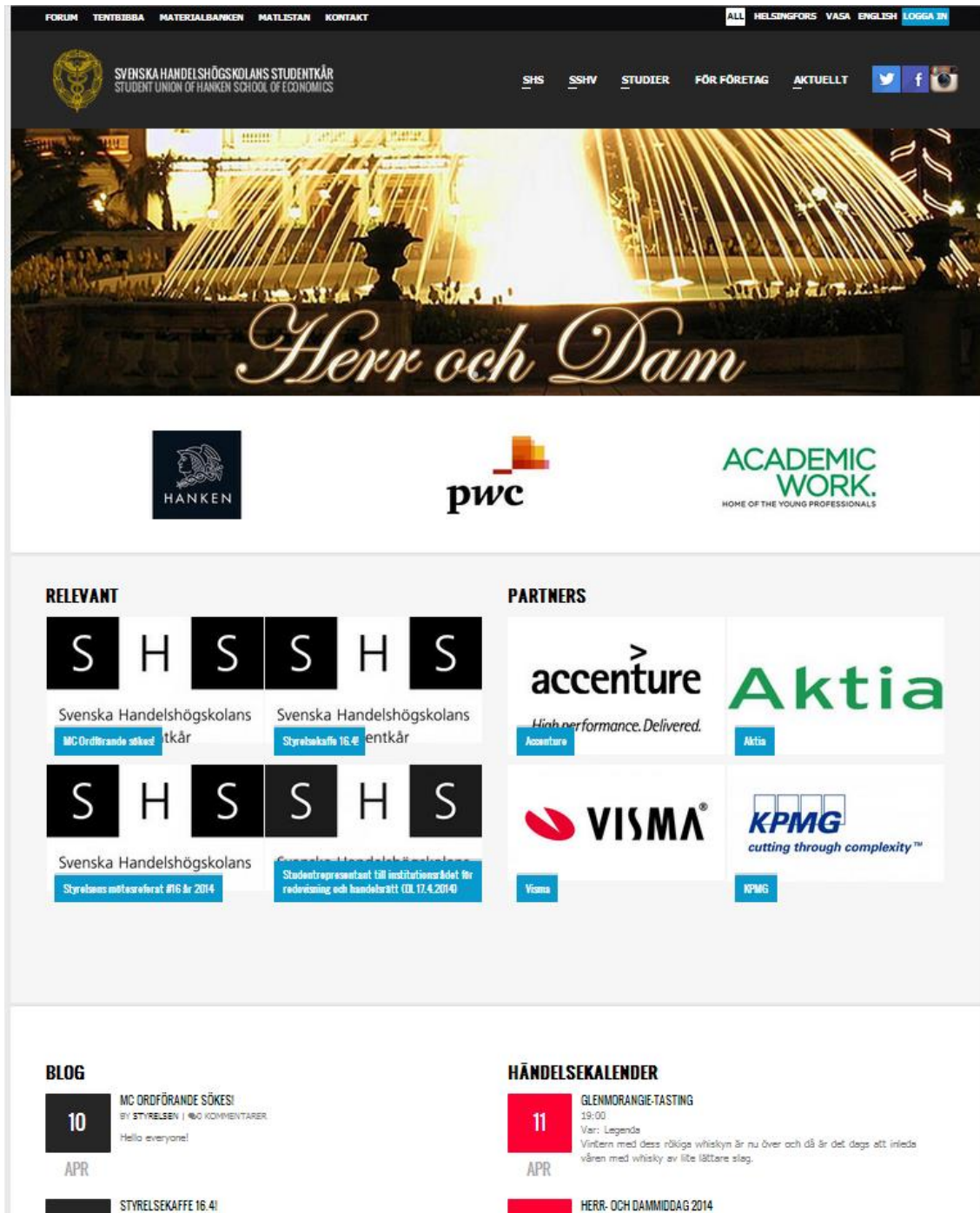
Figur 42. Val av campus

För funktionaliteten att välja vilket campus vyerna filtreras efter användes den kontribuerade modulen *Global Filter*. Modulen fungerar enligt principen *picked once, operates site-wide* (POOS) vilket betyder att valet sparas och används över hela hemsidan (Drupal.org 2014q). Valen av filtreringen sker via ett block som modulen erbjuder där en administrator kan specificera typen av värden som ska filtreras och hur alternativen ska presenteras. För ändamålet valdes att typen av filtervärde skulle vara nod fältet campus och typen av presentation radioknappar. För att filtret fungerade med radioknappar gavs förutom valet mellan Helsingfors och Vasa även ett alternativ för bägge campus.

För att en vy ska implementera filtreringen väljs *Content: campus* som kontextuellt filter i vyinställningarna. I filtrets inställningar väljs alternativet standardvärde då ett filteralternativ inte hittas och för typen av standardvärde väljs *Global filter (field or search terms)*. Efter detta kommer vyn att kolla vilket värde som är valt via det globala filtret och specificera resultatet efter valet.

6 DISKUSSION

Den färdiga produkten blev en fungerande webbplats där administratörer har möjligheten att producera material på både svenska och engelska (Figur 43). Sidan går att läsas på båda språken och listningssidornas material kan efter behov filtreras enligt önskat campus.



Figur 43. Framsidan för webbplatsen

6.1 Drupals temamöjligheter

En av målsättningarna var att redogöra hur ett Drupal-tema är uppbyggt och dess modifikationsmöjligheter. Drupals temamotor är ett mycket kraftfullt och flexibelt verktyg. Filsystemet är logiskt utlagt och de element som krävs för implementeringen kan summeras i en enda fil. För att göra ändringar till ett existerande Drupal-tema krävs inte mycket mer än att kopiera originalfilen till sin egna temakatalog och göra ändringar där de behövs.

Däremot kräver mer specifika ändringar i ett tema en del förkunskap i temamotorns logik och funktioner. Speciellt i fallet av omstrukturering via temafunktioner måste man ofta konsultera [Drupla.org](http://drupla.org) för att få reda på vilken funktion som egentligen är ansvarig för elementet i fråga. I fallet av huvudmenyns rullgardinsfunktionalitet var jag t.ex. tvungen att ersätta tre temafunktioner för att åstadkomma den slutliga modifikationen jag var ute efter. En betydlig del i modifikationen av teman går ut på att plocka ut de variabler som krävs för det värde som man vill komma åt. Sökandet av dessa variabler blir en ännu större börda i fall man saknar ett verktyg för att parsas systemets ibland minst sagt rikliga variabelhierarkier. Förkunskap av Drupals logik för strukturering av variabelhierarkin är här till stor hjälp.

Implementering av ett färdigt tema som ursprungligen inte byggts för Drupal kräver grundlig planering. En del av Colangine-temats funktionaliteter skalades bort eftersom en direkt implementation av dem skulle ha tagit en betydligt längre tid än att ha implementerat en färdig modul för ändamålet som använde ett annat jQuery-paket. *Showcase* var ett element som blev en aning klumpigt implementerat för listningsvyerna i och med att elementet ursprungligen planerats användas endast för en vy och nyttan för en dynamisk implementation skulle ha krävt mer tid än den skulle ha gjort nytta. Under projektets gång steg antalet vyer som använde *showcase*-elementet och en dynamisk format av elementet för vyer skulle ha minskat på antalet näst intill identiska mallar. Valet att flytta ut sidhuvudet och sidfoten ur *page*-mallen var också ämnad att förebygga onödigt arbete. Sidhuvudet blev modifierat ett flertal gånger under projektets gång och en del tid sparades i och med strukturförändringen. I slutet av projektet stod det dock klart att antalet *page*-mallar var lågt och konstruktionen medförde inte en förenkling av den skala den kunde ha åstadkommit.

6.2 Innehållstrukturen

Precis som med temats tillämpningar är skapandet av innehållsstrukturen något som kräver mycket planering före utförandet. Genom att återanvända fält för flera innehållstyper hålls kan variablerna för sid- och innehållsmallarna använda sig av samma variabler och fält kan mer koncist placeras ut på en sidmall.

Ändringar av ett fält eller likartad komponent i innehållsstrukturen är ingen större korrigering förutsatt att inga inlägg där fältet använts sparats. I en del fall måste fältet eller innehållskomponenten raderas i sin helhet och byggas om med följden av att all information som tidigare matats in via fältet måste raderas i processen.

Vid sparandet av ett fält skapas automatiskt ett maskinläsbart namn, utan nordiska tecken och specialtecken, för fältet. Denna funktion sparar en hel del tid då fältets namn är på engelska men blir snabbt en fallgrop då innehållens formulär skapas för ett annat språk. I användargränssnittet spelar maskin namnet ingen större roll men på kodsidan där allting i regel är på engelska blir de icke engelska variablerna förvirrande och ibland oläsliga.

6.3 Slutsats

Drupal lämpade sig mycket väl för denna typ av webbplats. Funktionaliteten att skapa innehållstyper och kategorisering medförde att själva innehållet på sidan kunde börja införas tidigt i projektet. Följden av att i ett tidigt skede ha innehåll till förfogande är att listningsvyerna också snabbt kan börja byggas och testas. För flerspråkigheten erbjöd Drupal 7 en handfull kontribuerade moduler som alla gjorde sin del av översättandet med slutresultatet att effektiv tvåspråkighet går att åstadkomma på en Drupal-webbplats men alternativen är flera och ingen modul erbjuder en total lösning. Ibruktagandet av ett tema som inte var byggt för Drupal var i sin grundimplementation ingen svår process. De större sidkomponenterna som regioner för Drupal-innehåll var enkla att tillämpa på en dynamisk nivå. Däremot ledde tillämpningen av mindre element ofta till snabba lösningar som i slutändan kunde leda till att komponenter blev mindre dynamiska på långsikt.

KÄLLOR

Aschan, Marcus. 2011, *Utveckling av tema för Drupal*, slutarbete.

BuiltWith.com 2014, *CSM Usage Statistics*, [Statistik], Hämtat 18.3.2014, <http://trends.builtwith.com/cms>

Drupal.org 2014a, *About Drupal*, [Artikel], Hämtad 17.3.2014, <https://drupal.org/about>

Drupal.org 2014b, *About Nodes*, [Artikel], Hämtad 18.3.2014, <https://drupal.org/documentation/modules/node>

Drupal.org 2014c, *Working with content types and fields (Drupal 7 and later)*, [Dokumentation], Hämtat 18.3.2014, <https://drupal.org/documentation/modules/field-ui>

Drupal.org 2014d, *Working with blocks (content in regions)*, [Dokumentation], Hämtad 28.4.2014, <https://drupal.org/documentation/modules/block>

Drupal.org 2014e, *Working with Menus*, [Dokumentation], Hämtad 18.3.2014] <https://drupal.org/documentation/modules/menu>

Drupal.org 2014f, *About taxonomies*, [Artikel], Hämtad 28.4.2014, <https://drupal.org/node/774892>

Drupal.org 2014g, *Module developer's guide*, [Artikel], Hämtad 18.3.2014, <https://drupal.org/developing/modules>

Drupal.org 2014h, *Overview of theme files*, [Artikel], hämtad 25.2.2014, <https://drupal.org/node/171194>

Drupal.org 2014i, *Writing theme .info files*, [Artikel], Hämtad 25.2.2014, <https://drupal.org/node/171205>

Drupal.org 2014j, *Convert any website layout or template into a Drupal theme - easily!*, [Artikel], Hämtad 25.2.2014 <https://drupal.org/node/313510>

Drupal.org 2014k, *The Drupal overview*, [Artikel], Hämtad 11.3.2014, <https://drupal.org/getting-started/before/overview>

Drupal.org 2014l, *Customize the front page template*, [Artikel], Hämtad 11.3.2014, <https://drupal.org/node/317461>

Drupal.org 2014m, *Forum: create threaded discussions*, [Dokumentation], Hämtad 1.4.2014, <https://drupal.org/documentation/modules/forum>

- Drupal.org 2014n, *function l*, [Dokumentation], Hämtad 4.4.2014, <https://api.drupal.org/api/drupal/includes!common.inc/function/l/7>
- Drupal.org 2014o, *Translate content on field level with Entity translation*, [Artikel], Hämtad 2.4.2014, <https://drupal.org/node/1280632>
- Drupal.org 2014p, *function t*, [Dokumentation], Hämtad 2.4.2014, <https://api.drupal.org/api/drupal/includes!bootstrap.inc/function/t/7>
- Drupal.org 2014q, *Views Global Filter*, [Artikel], Hämtad 4.4.2014, https://drupal.org/project/global_filter
- Duckett, Jon. 2011, *HTML & CSS: Design and Build Websites*, Indianapolis: John Wiley & Sons, 513 s.
- Kumar, Karthik. 2012, *Drupal 7 Theming Cookbook*, 2 uppl., Birmingham: Packt Publishing Ltd., 364 s.
- Melacon, Benjamin; Micka, Allie; Scavarda, Amye; Somers, Bojhan; Luisi, Jacine; Negyesi, Karloy; Scholtzen, Roy; Szarma, Ryan; Boyer, Sam; Corlosquet, Stephane; Hakimzadeh, Dan; Dolin, Kasey; Freudenberg, Stefan, 2011, *The definitive guide to Drupal 7*, Apress, 1112 s.
- Rambeck, Nathan. 2014, *Organizing your theme files in Drupal*, [Blog], Hämtat 5.3.2014, <http://nathan.rambeck.org/blog/17-organizing-your-theme-files-drupal>
- ThemeForest.net 2014, *Colangine - Creative Flat HTML5 Template*, [Produktsida], Hämtat 11.4.2014, <http://themeforest.net/item/colangine-creative-flat-html5-template/5214550>